

MP1630B

Remote Control

Operation Manual

First Edition

To ensure that the equipment is used safely, read the "For Safety" in the MP1630B Digital Data Analyzer Operation Manual first.

Keep this manual with the equipment.

Measuring Instruments Division
Measurement Group
ANRITSU CORPORATION

Document No. : M-W1443AE-1.0

MP1630B
Operation Manual
Remote Control Operation Manual
31 August 1998(First Edition)

Copyright © 1998, ANRITSU CORPORATION

All right reserved. No part on this manual may be reproduced without the prior written permission of the publisher.

The contents of this manual may be changed without prior notice.

Printed in Japan

Trademark

Microsoft, Visual Basic, MS-DOS, Windows are registered trademark of Microsoft Corporation US in US and other countries.



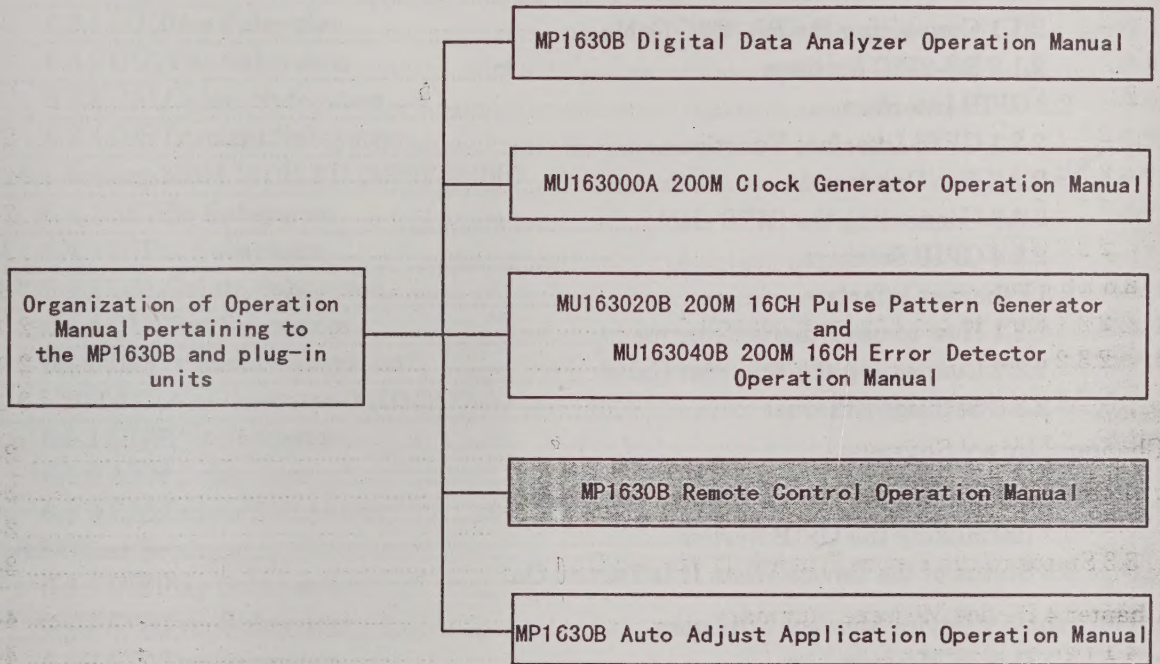
Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_06318

Introduction

Organization of Operation Manuals

The MP1630B Digital Data Analyzer is a measuring instrument designed to house plug-in units. Operation Manuals are divided into five volumes pertaining to the MP1630B Digital Data Analyzer itself and the individual plug-in units. Read these Operation Manual according to the intended applications.



- **MP1630B Digital Data Analyzer Operation Manual**
Gives an overview of the MP1630B and describes its setup steps, its panels, the relevant standards, performance details, and how to operate the unit.
- **MU163000A 200M Clock Generator Operation Manual**
Gives an overview of the MU163000A 200M clock generator unit and describes the relevant standards, performance details, and how to operate the unit.
- **MU163020B 200M 16CH PPG&MU163040B 200M 16CH ED Operation Manual**
Gives an overview of the 200M 16CH PPG and ED units and describes the relevant standards, performance details, and how to operate the units.
- **MP1630B Remote Control Operation Manual**
Explains the concepts of external interface (RS-232C, GPIB option,ETHERNET option) control, and also contains sample programs.
- **MP1630B Auto Adjust Application Operation Manual**
Explains how to operate the relevant to the MP1630B Auto Adjust Application Operation Manual

Contents

Introduction	I
Chapter 1 Overview	1-1
1.1 Interface Functions.....	1-2
1.2 Examples of System Upgrading.....	1-3
Chapter 2 Connecting and Setting Up Interfaces.....	2-1
2.1 RS-232C Interface.....	2-2
2.1.1 Connecting the RS-232C Cable	2-2
2.1.2 RS-232C Settings	2-3
2.2 GPIB Interface	2-5
2.2.1 GPIB Interface Functions.....	2-5
2.2.2 Bus Commands.....	2-6
2.2.3 Connecting the GPIB Cable.....	2-7
2.2.4 GPIB Settings.....	2-8
2.3 Ethernet Interface	2-10
2.3.1 How to use Ethernet control.....	2-10
2.3.2 Connecting the Ethernet Cable.....	2-10
2.3.3 Settings Ethernet.....	2-11
Chapter 3 Initial Settings.....	3-1
3.1 Initializing the Device	3-2
3.2 Initializing the GPIB System.....	3-3
3.3 Status of the Device When It Is Turned On.....	3-6
Chapter 4 Device Message Summary	4-1
4.1 Device Message List.....	4-2
4.2 Listener Input Format.....	4-3
4.3 Talker Output Format.....	4-7
4.4 Common Commands.....	4-9
4.5 SCPI Summary	4-14
Chapter 5 Status Reporting.....	5-1
5.1 MP1630B Status Register Configuration	5-2
5.2 IEEE488.2-prescribed Status Registers.....	5-4
5.3 SCPI-prescribed Status Registers	5-6
5.4 Device-specific Status Registers.....	5-8
5.5 Reading and Writing Status Registers.....	5-15
5.6 Clearing and Resetting Status Registers.....	5-16

Chapter 6 Device Message Details	6-1
6.1 Command List	6-2
6.1.1 Subsystem Configuration	6-2
6.1.2 Command Trees	6-3
6.2 Before Reading Detailed Command Descriptions	6-28
6.2.1 Tips on Transmitting Commands	6-28
6.2.2 Buffer Sizes	6-29
6.2.3 How to Read Detailed Command Descriptions	6-30
6.3 Setting 200M 16CH PPG (MP163020B)	6-31
6.3.1 SOURce Subsystem	6-31
6.3.2 OUTPut Subsystem	6-57
6.3.3 CALCulate Subsystem	6-69
6.3.4 INSTRument Subsystem	6-70
6.4 Setting 200M 16CH ED (MP163040B)	6-71
6.4.1 SENSE Subsystem	6-71
6.4.2 INPut Subsystem	6-117
6.4.3 CALCulate Subsystem	6-121
6.4.4 OUTPut Subsystem	6-134
6.4.5 INSTRument Subsystem	6-136
6.5 200M Clock Generator (MU163000A)	6-137
6.5.1 OUTPut Subsystem	6-137
6.5.2 ARM Subsystem	6-139
6.5.3 CALCulate Subsystem	6-140
6.6 Other Settings	6-141
6.6.1 DISPlay Subsystem	6-141
6.6.2 SYSTem Subsystem	6-165
6.6.3 STATus Subsystem	6-178
Chapter 7 Sample Programs	7-1
7.1 Programming Environment	7-2
7.2 Program Descriptions	7-3
Appendix	
Appendix A SCPI Conformance Information	A-1
Appendix B SCPI Error Messages	B-1
Appendix C Command List	C-1

Chapter 1 Overview

1.1 Interface Functions

The MP1630B digital data analyzer (hereafter simply called this instrument) comes complete with an RS-232C interface. It optionally supports the GPIB and Ethernet interfaces.

The RS-232C, GPIB, and Ethernet interfaces can all be supported at the same time.

The control commands provided by the instrument conform to the specifications of SCPI (Standard Commands for Programmable Instruments), so that the same repertoire of commands is operable regardless of the interface type.

- RS-232C interface
The instrument supports an RS-232C (COM1) interface as a standard feature. A controller connected to the instrument via the RS-232C interface controls its operation.
- GPIB interface (OPT01)
The instrument supports a GPIB interface conforming to IEEE std488.1-1987. A controller connected to the instrument via the GPIB interface controls its operation.
- Ethernet interface (OPT02)
The instrument supports an Ethernet interface conforming to IEEE 802.3. A controller connected to the instrument via the Ethernet interface controls its operation.

This instrument provides the following interface functions:

- Control of all the functions, except for the following:
 - Power switch and Local key
 - Interface settings, such as the RS-232C baud rate
 - Dialog box opening and closing, and onscreen operations, such as pattern editor cursor positioning
- Reading of all setup conditions and display images
- Implementation of an automatic measurement system in conjunction with a personal computer
- Interrupt facilities and serial polling (if the GPIB interface option is installed)

1.2 Examples of System Upgrading

Examples of systems upgraded by using the interfaces supported by the instrument are shown below.

- Host computer control

A host computer can be connected to the instrument to make up an automatic measurement system.

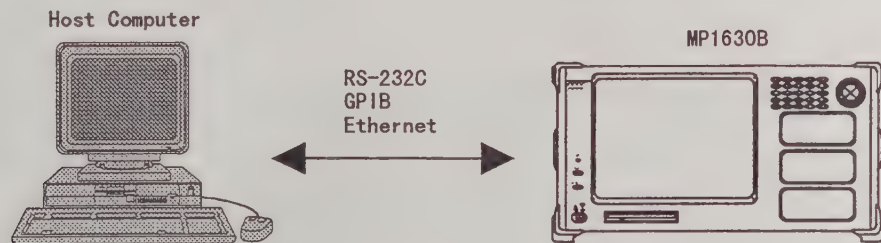


Figure 1-1 Host Computer Control

1-1
1-2
1-3
1-4

1-5
1-6
1-7
1-8
1-9

1-10

Chapter 2 Connecting and Setting Up Interfaces

2.1 RS-232C Interface

2.1.1 Connecting the RS-232C Cable

The RS-232C cable connector is found on the rear panel. Be sure to connect the RS-232C cable to the RS-232C cable connector before turning on the power.

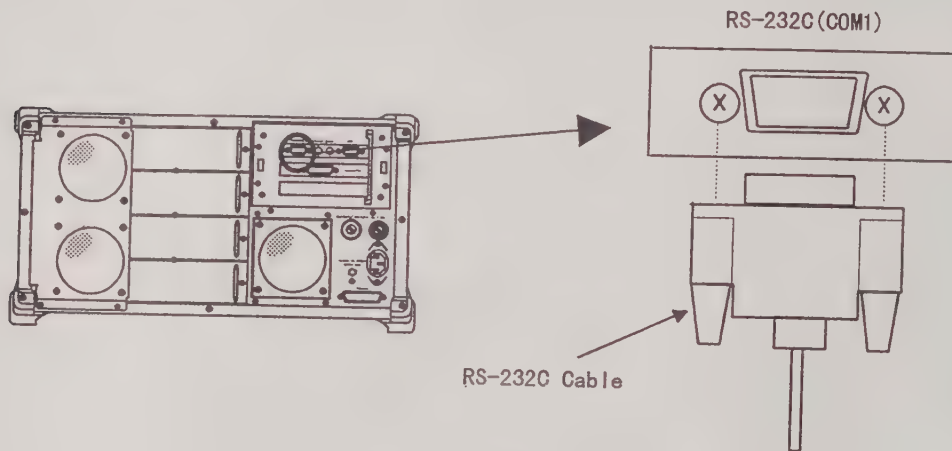


Figure 2-1 Connecting the RS-232C Cable

2.1.2 RS-232C Settings

If RS-232C is used as a remote control interface, set the instrument in the local mode first and complete the settings listed below from the instrument's screen. These settings can be altered at any time while the instrument is in the local mode.

Table 2-1 RS-232C Settings

Setting	Setup parameter	Setup range	Initial value
Remote control interface setting	Remote	RS-232C (COM1) GPIB (if OPT01 is installed) Ethernet (if OPT02 is installed) None	RS-232C(COM1)
Baud rate setting	Speed	110, 300, 600, 1200, 2400, 4800, 9600 9600, 19200	
Parity setting	Parity	None, Even, Odd	None
Character length setting	Data Bit	8bits, 7bits	8bits
Stop bit length setting	Stop Bit	2bits, 1bit	1bits
Flow control setting	Flow Control	Xon/Xoff, Hardware, None	Hardware

These settings are entered from the System:System panel. Follow these steps:

1. Open the System:System panel.

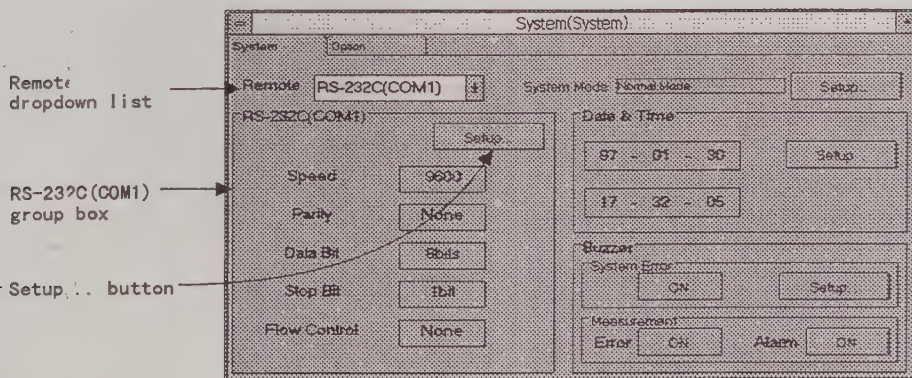


Figure 2-2 System:System Panel

2. Select RS-232C (COM1) from the Remote dropdown list. The RS-232C (COM1) group box will then appear on the System:System panel.

3. Click the Setup... button in the RS-232C (COM1) group box.

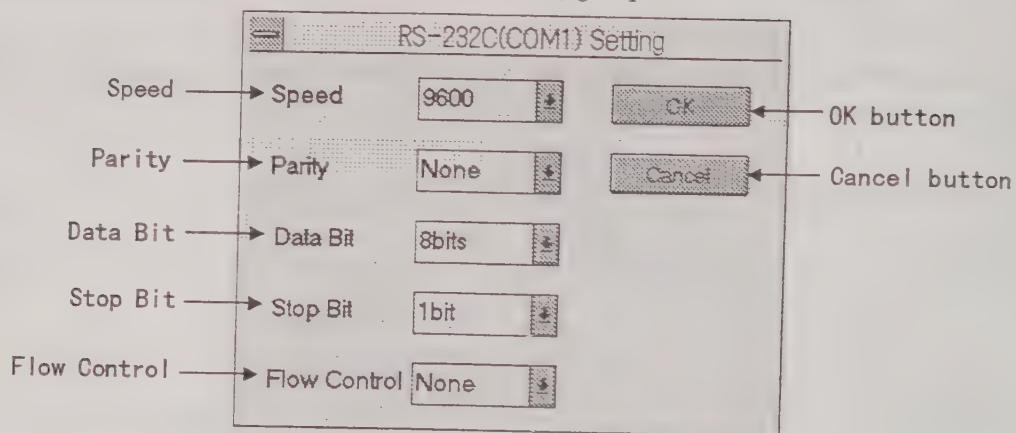


Figure 2-3 RS-232C (COM1) Dialog Box

4. Set the individual items to suit your environment.
5. The RS-232C setup procedure is now complete. Click the OK button to enable the settings, or click the Cancel button to abandon them.

2.2 GPIB Interface

2.2.1 GPIB Interface Functions

Out of the GPIB interface functions specified by IEEE488.1, IEEE488.2 prescribes minimum subsets that are required of measuring instruments to meet at least system application needs. The table below lists the subsets in the Code column that are provided by the GPIB interface supported by this instrument.

Table 2-2 GPIB Interface Functions

Code	Interface functions	IEEE488.2 specification
SH1	All source handshaking functions available. Data transmission timing provided.	All functions available as standard features.
AH1	All acceptor handshaking functions available. Data reception timing provided.	All functions available as standard features.
T6	Basic talker and serial polling functions available, but no talk-only function. MLA-directed talker clear function available.	Devices must have any one of the subsets T5, T6, TE5, and TE6.
L4	Basic listener and listen-only functions available. MLA-directed listener clear function available.	Devices must have any one of the subsets L3, L4, LE3, and LE4.
SR1	All service request and status byte functions available.	All functions available as standard features.
RL1	All remote/local functions and local lockout function available.	RL0 (no function) or RL1 (all functions).
PP0	No parallel polling function available.	PP0 (no function) or PP1 (all functions).
DC1	All device clear functions available	All functions available as standard features.
DT1	All device trigger functions available	DT0 (no function) or DT1 (all functions).
C0	No control functions.	C0 (no function) or C4 and C5, or any one of C7, C9, and C11.

2.2.2 Bus Commands

Bus commands are internal interface communications that are transmitted when the GPIB bus is in the command mode (the ATN line is low).

GPIB used as a remote control interface allows the instrument to be controlled by issuing bus commands from the controller.

The table below lists the bus commands available.

Table 2-3 Bus Commands

Bus command	Operation
DCL(Device clear)	Initializes message switching by all the devices on the GPIB bus.
SDC(Selected Device Clear)	Initializes message switching by an addressed device on the GPIB bus. Similar in function to DCL.
GET(Group Execute Trigger)	Executes the same function as the function that is executed by pressing the Start key.
IFC(Interface Clear)	Initializes the interface.

2.2.3 Connecting the GPIB Cable

If the GPIB option (OPT01) is installed, a connector is available on the rear panel to receive a GPIB cable. (The GPIB option is a factory option.) Be sure to connect the GPIB cable to the GPIB cable connector before turning on the power.

Up to 15 devices can be connected to a single system, including a controller. The following limitations are placed on the cable lengths:

- Sum total of the cable lengths ≤ 20 m
- Device-to-device cable length ≤ 4 m

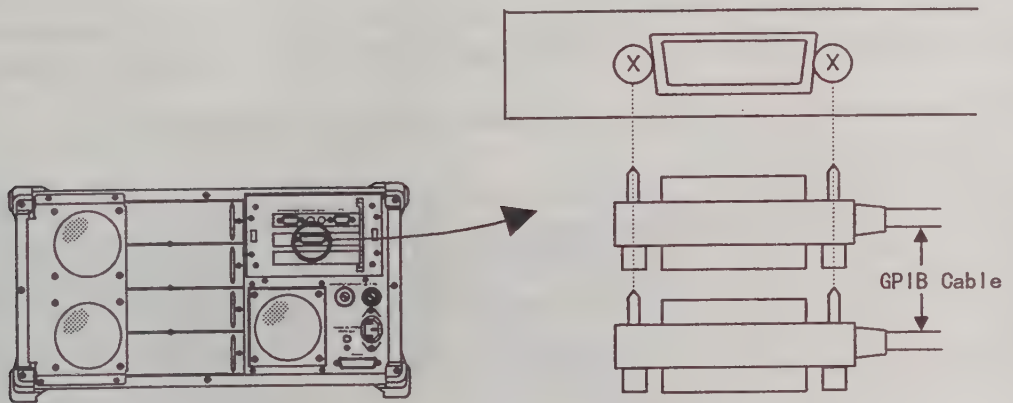


Figure 2-4 Connecting the GPIB Cable

2.2.4 GPIB Settings

If GPIB is used as a remote control interface, set the instrument in the local mode first and complete the settings listed below from the instrument's screen. These settings can be altered at any time while the instrument is in the local mode.

Table 2-4 GPIB Settings

Setting	Setup parameter	Setup range	Initial value
Remote control interface setting	Remote	RS-232C (COM1) GPIB (if OPT01 is installed) Ethernet (if OPT02 is installed) None	RS-232C(COM1)
GPIB address setting	GPIB Address	0~30/Step:1	1

These settings are entered from the System:System panel. Follow these steps:

1. Open the System:System panel.

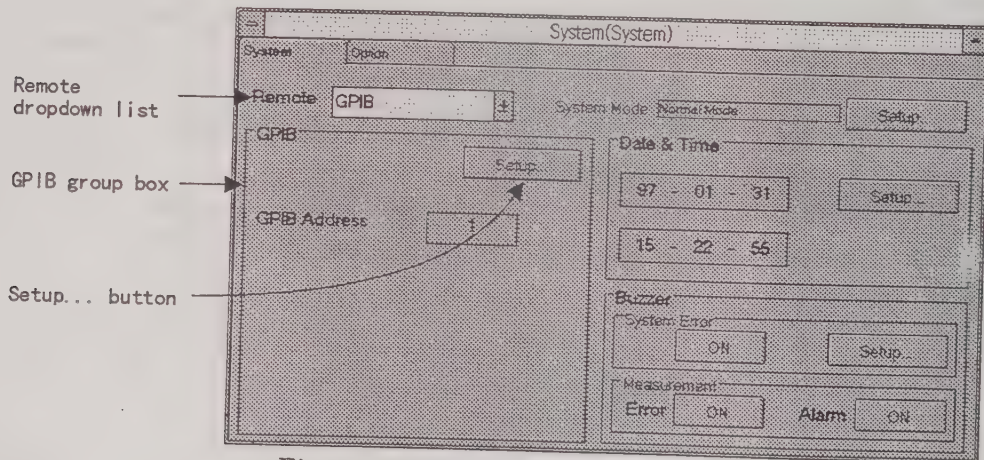


Figure 2-5 System:System Panel

2. Select GPIB from the Remote dropdown list. The GPIB group box will then appear on the System:System panel.

3. Click the Setup... button in the GPIB group box.

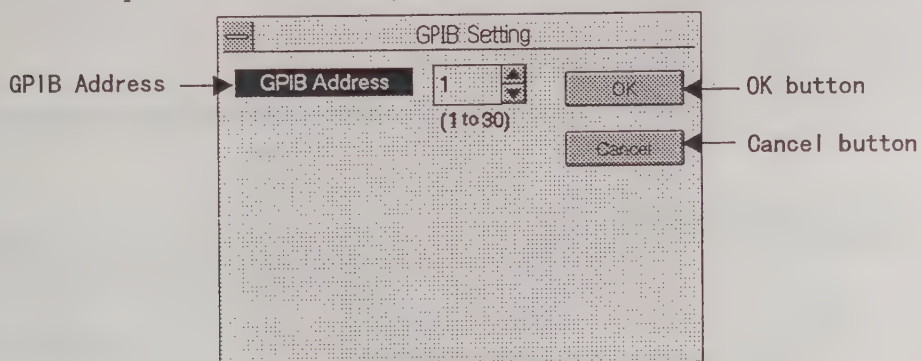


Figure 2-6 GPIB Dialog Box

4. Set the individual items to suit your environment.
5. The GPIB setup procedure is now complete. Click the OK button to enable the settings, or click the Cancel button to discard them.

Note

The GPIB address setting of this instrument is stored in backup memory even when it is turned off. The instrument, when turned on, lets an application set its own address. While the instrument is turned on, its address can be altered from the screen at any time unless it is in the remote mode. A change to the address takes effect when a change dialog box is closed by clicking the OK button.

2.3 Ethernet Interface

2.3.1 How to use Ethernet control

When using Ethernet as the remote interface, MP1630B operates as a server. Since it uses port No. 5001, designate this port No. for connection on the client.

Note

When using Ethernet as the remote interface, you should link the Analyzer to the TCP/IP network.

For further information on linking to the network, see the MP1630B Digital Data Analyzer Ethernet Option Operation Manual.

2.3.2 Connecting the Ethernet Cable

If the Ethernet option (OPT02) is installed, a connector is available on the rear panel to receive a Ethernet cable(10BASET Twist pair cable). (The Ethernet option is a factory option.) Be sure to connect the Ethernet cable to the Ethernet cable connector before turning on the power. Insert a RJ-45 jack in the correct direction until you hear a click sound.

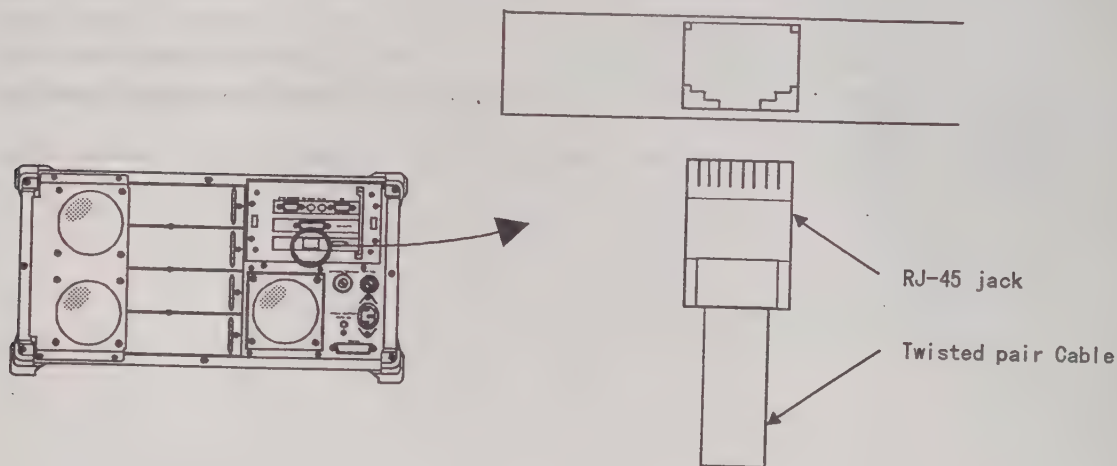


Figure 2-7 Connecting the Ethernet Cable

2.3.3 Settings Ethernet

If Ethernet is used as a remote control interface, set the instrument in the local mode first and complete the settings listed below from the instrument's screen. These settings can be altered at any time while the instrument is in the local mode.

Table 2-5 Ethernet Settings

Setting	Setup parameter	Setup range	Initial value
Remote control interface setting	Remote	RS-232C (COM1) GPIB (if OPT01 is installed) Ethernet (if OPT02 is installed) None	RS-232C(COM1)

These settings are entered from the System:System panel. Follow these steps:

1. Open the System:System panel.

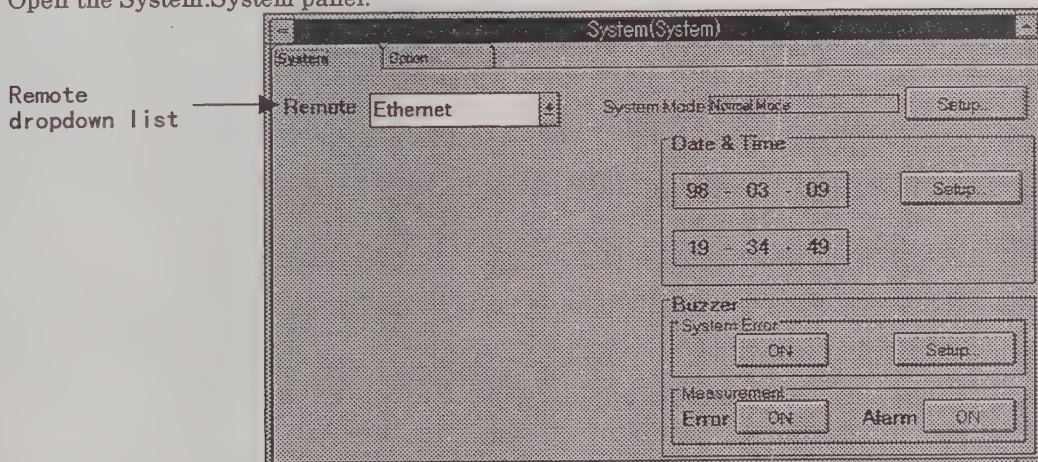


Figure 2-8 System:System Panel

2. Select Ethernet from the Remote dropdown list.

Chapter 3 Initial Settings

3.1 Initializing the Device

This instrument uses the following two commands to initialize itself as a device. These commands achieve level 3 initialization (device initialization) prescribed by IEEE488.2.

- *RST common command
- :SYSTem:MEMory:INITialize command

■ Initializing a device with the *RST common command

An IEEE488.2 common command, *RST is operable regardless of the interface type.

The execution of the *RST command puts a device into the following conditions:

1. The device-specific functions are reset to a given known status, regardless of the history of their usage. This instrument is reset to their factory defaults.
2. The device is put into the OCIS (Operation Complete Command Idle State).
3. The device is put into the OQIS (Operation Complete Query Idle State).

Note

The execution of the *RST command does not influence the following:

1. IEEE488.2 interface status
2. Device address
3. Output queue
4. Service request enable register
5. Standard event status enable register
6. Power-on-status-clear flag setting
7. Configuration data affecting device specifications
8. Macros defined with the DMC command
9. Response messages to PUD queries
10. Response messages to RDT queries

(Among these, 6 and 8 through 10 are not supported by this instrument.)

■ Initializing a device with the :SYSTem:MEMory:INITialize command

A device-specific command, the SYSTem:MEMory:INITialize command is operable regardless of the interface type.

The execution of the :SYSTem:MEMory:INITialize command initializes the device settings to their factory defaults.

Note

The execution of this command does not initialize the interface settings (such as the RS-232C baud rate).

3.2 Initializing the GPIB System

If GPIB is used as a remote control interface, the GPIB system is initialized at the levels described below, in addition to level 3 (device initialization) described earlier.

The table below defines bus initialization (level 1) and message initialization (level 2).

Table 3-1 Levels of GPIB System Initialization

Level	Kind of initialization	Summary
1	Bus initialization	Initializes all the interface functions connected to the bus with an IFC message from the controller.
2	Message initialization	Initializes message switching by all the devices on the GPIB bus with the GPIB bus command DCL or initializes message switching by an addressed device on the GPIB bus with the GPIB bus command SDC, and disables the function of reporting the completion of operations to the controller.
3	Device initialization	Resets the device-specific functions to given known status with the *RST or SYSTem:MEMory:INITialize command, regardless of the history of their usage.

■ Initializing the bus with an IFC statement

An IFC statement initializes all the device functions connected to the GPIB bus line.

Initializing interface functions consists in resetting the interface functions of a device that have been set by the controller to their initial status.

The table below lists the interface functions that are initialized by an IFC message. In the table, the functions marked by ○ are initialized in their entirety, while those marked by △ are initialized in part. The execution of this command does not influence the operating status of the device.

Table 3-2 Changes in the Status of Interface Functions Effected by the IFC Message

No	Function	Symbol	Initialization by IFC
1	Source handshaking	SH	○
2	Acceptor handshaking	AH	○
3	Talker or extended listener	T or TE	○
4	Listener or extended listener	L or LT	○
5	Service request	SR	△
6	Remote local	RL	
7	Parallel poll	PP	
8	Device clear	DC	
9	Device trigger	DT	
10	Controller	C	○

The IFC message brings about the following changes in device status:

Table 3-3 Changes in Device Status Effected by the IFC Message

Talker listener	All talkers and all listeners enter an idle state (TIDS, LIDS) within 100 μ s.
Controller	The controller enters the idle state CIDS (Control IDle State) unless it is in the active state (SACS: System control ACtive State).
Service requesting Device	The state of the device transmitting an SRQ message to the controller, or the state in which the SRQ line is held low by the device, is not reset, but the state of the controller having all the devices on the system bus in the serial poll mode is reset.
Device in the remote mode	A device currently in the remote mode is never reset from the remote mode by the IFC message.

■ Initializing message switching with the DCL and SDC bus commands

The DCL and SDC bus commands initialize message switching by all the devices or by an addressed device on the GPIB bus.

The table below lists the items of message switching that are initialized by these commands.

Table 3-4 Message Switching Items That Are Initialized

Input buffer and output queue	Cleared.
Syntax analysis, execution control, and response generation parts	Reset.
Device commands, including *RST	Clears any command that interferes with the execution of these commands.
Parameter program messages	Abandons all the commands and queries whose execution has been deferred because of parameters.
*OPC command processing	The device is put into the OCIS (Operation Complete Command Idle State). As a result, the operation complete bit cannot be set in the standard event status enable register.
OPC? query processing	The device is put into the OQIS (Operation Complete Query Idle State). As a result, operation complete bit 1 cannot be set in the output queue. The MAV bit is cleared.
Device functions	All functions pertaining to message switching are put into an idle state. The device continues to wait for a message from the controller.

3.3 Status of the Device When It Is Turned On

The instrument, when turned on, enters the following conditions:

1. The instrument comes up in the status in which it was when it was last turned off.
 2. Both the input and output queues are cleared.
 3. The syntax analysis, execution control, and response generation parts are reset.
 4. The device is put into the OCIS state.
 5. The device is put into the OQIS state.
 6. All the event status enable registers are cleared. Events are logged after the clearing.
- (*PSC command is not supported by this instrument.)

Chapter 4 Device Message Summary

4.1 Device Message List

Device messages are data messages that are transmitted between the controller and device via the system interface. They are grouped into program messages and response messages.

The program messages are ASCII data messages that are transmitted from the controller to the device, while the response messages are ASCII data messages that are transmitted from the device to the controller.

The table below breaks down the program messages and response messages.

Table 4-1 Device Message List

Program messages	Response messages
• Program commands • Device-specific commands • (See Chapter 6 for more details.) • IEEE488.2 common commands • (See Section 4.4 for more details.) • Program queries (See Chapter 6 for more details.)	• Status messages (See Chapter 5 for more details.) • Response messages (See Chapter 6 for more details.)



Figure 4-1 Device Message List

The messages mentioned above are transmitted by way of I/O buffers in the device. The I/O buffers are briefly described below.

Table 4-2 I/O Buffers

Input buffer	Output buffer
FIFO (first-in first-out) memory area in which DABs (program messages and query messages) are stored temporarily. This instrument provides an input buffer size of 16 Kbytes.	FIFO queue memory area. All DABs that are directed from the device to the controller are stored in this area until the controller finishes reading them. This instrument provides an output buffer size of 16 Kbytes.

4.2 Listener Input Format

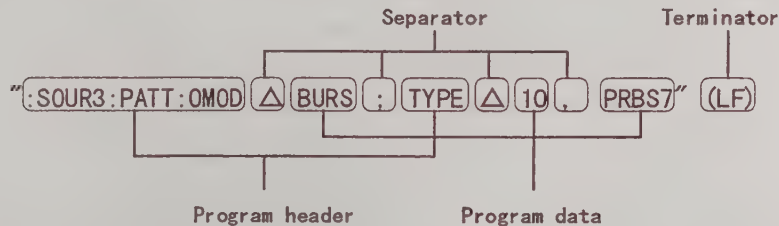
The control commands provided by the instrument conform to the specifications of IEEE488.2, as well as SCPI. IEEE488.2 defines a program message as a sequence of functional elements.

A functional element is the smallest level of function representation. Functional elements are broadly divided into separators, terminators, program headers, and program data.

Example: When the program message

":SOUR3:PATT:OMOD BURS;TYPE 10,PRBS7" (LF)

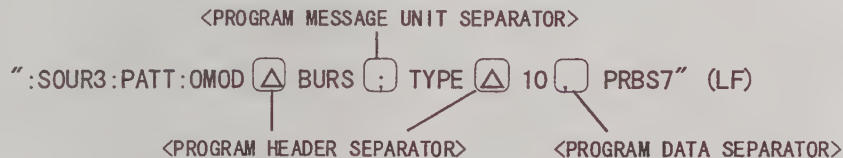
is transmitted from the controller to the device (command that sets the burst transmitted pattern output mode and sets PRBS7 as a CH10 pattern).



This section breaks down the instrument's command format into these four kinds of functional elements and describes each. (In the explanations, Δ denotes a space.)

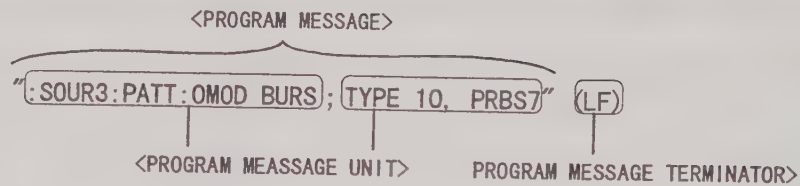
■ Separators

Functional elements are available as separators as described below.



Functional element	Explanation
<PROGRAM MESSAGE UNIT SEPARATOR>	Separates multiple <PROGRAM MESSAGE UNIT> elements from one another. Represented by 0, one, or more spaces and a semicolon. Example: One space + Semicolon :SOUR3:PATT:OMOD BURS Δ ; TYPE 1,PRBS7
<PROGRAM DATA SEPARATOR>	Separates multiple <PROGRAM DATA> elements from one another. Represented by 0, one, or more spaces, a comma, and 0, one, or more spaces. Example: Comma only :SOUR3:PATT:TYPE 1, PRBS7
<PROGRAM HEADER SEPARATOR>	Separates a program header and a <PROGRAM DATA> element from each other. Represented by one or more spaces. Example: One space :SOUR3:PATT:OMOD Δ BURS

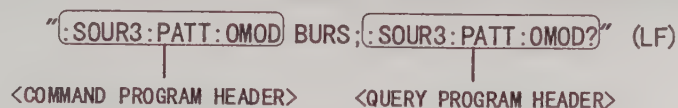
■ Terminator



Functional element	Explanation
<PROGRAM MESSAGE TERMINATOR>	Attached at the end of a <PROGRAM MESSAGE> to terminate a sequence of one or more <PROGRAM MESSAGE UNIT> elements. Represented by one or more spaces, (CR + LF or LF) and (EOI ON or EOI OFF).

■ Program headers

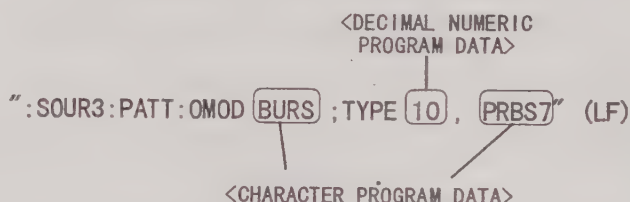
Two functional elements are available as program headers as described below.



Functional element	Explanation
<COMMAND PROGRAM HEADER>	A command used by the controller to control the device. The <COMMAND PROGRAM HEADER> supported by this instrument is classified into one of two alternative coded syntaxes: <common command program header> and <compound command program header>. <common command program header> IEEE488.2 common commands use this syntax. Example: *RST <compound command program header> Device-specific commands (SCPI) use this syntax. Example: <u>:SOUR3:PATT:OMOD</u> BURS
<QUERY PROGRAM HEADER>	A query command is a command for sending a query from the controller to the device so that the controller can receive a response message from it. The header is terminated by a query indicator (?). The <QUERY PROGRAM HEADER> supported by this instrument is classified into one of two alternative coded syntaxes: <common query program header> and <compound query program header>. <common query program header> IEEE488.2 common query commands use this syntax. Example: *IDN? <compound query program header> Device-specific query commands (SCPI) use this syntax. Example: <u>:SOUR3:PATT:OMOD?</u>

■ Program data

Program data supported by this instrument is grouped into six functional elements as described below. In detailed descriptions of commands, the IEEE488.2-prescribed program data types associated with device-specific parameter types are given.



Functional element	Explanation
<CHARACTER PROGRAM DATA>	A short character string associated with a parameter function.
<DECIMAL NUMERIC PROGRAM DATA>	Represents a decimal integer or real number. This instrument uses integers and fixed-decimal point numbers: Examples: Integers 005, +5, 5, +5△ Fixed-decimal point numbers: 12.345, .05, +0.05, 12.
<NON-DECIMAL NUMERIC PROGRAM DATA>	Represents a non-decimal numeric value. This instrument uses hexadecimal numbers, which are expressed by #H+{A through F, a through f, 0 through 9). Examples: #HABC, #H123
<STRING PROGRAM DATA>	Represents a character string enclosed in double or single quotation marks. Examples: "ABCD", '1234'
<ARBITRARY BLOCK PROGRAM DATA>	Used to transmit 8-bit binary data. #YYYYYY<binary> X :Number of digits of YYYYYY YYYYYY :Byte length of <binary> <binary> :Binary data Examples: #14<DAB><DAB><DAB><DAB> <DAB> : Data byte
<BOOLEAN PROGRAM DATA>	Program data defined by SCPI representing a theoretical value of either ON or 1, or OFF or 0.

Note

With SCPI instruments, the character strings represented by <CHARACTER PROGRAM DATA> and <STRING PROGRAM DATA> have long and short forms. A short form is an abbreviation of the corresponding long form. In command descriptions, short forms represent the uppercase portion of a character string.

A short form and the corresponding long form are interpreted to refer to the same function. Though distinction is made between upper and lowercase letters in the explanations, no such distinction is made.

Examples : :SOUR3:PATT:OMOD BURSt (BURSt is a long form.)
:SOUR3:PATT:OMOD BURS (BURS is a short form.)

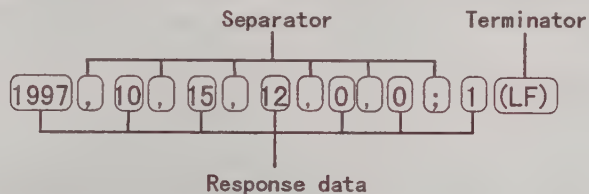
4.3 Talker Output Format

The control commands provided by the instrument conform to the specifications of IEEE488.2, as well as SCPI. IEEE488.2 defines a response message as a sequence of functional elements as it defines a program message.

Functional elements are broadly divided into separators, terminators, response headers, and response data.

Example: Response 1997,10,15,12,0,0;1 to the command querying the

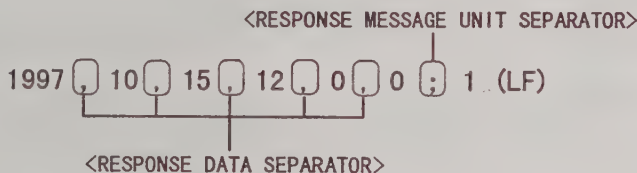
:SENSe4:MEASure:EALarm:BTIMe:STARt?,SET?BLOC:SIZE?



Since response headers are not included in the response messages that are returned by this instrument, this section breaks down the instrument's command format into these three kinds of functional elements and describes each. (In the explanations, Δ denotes a space.)

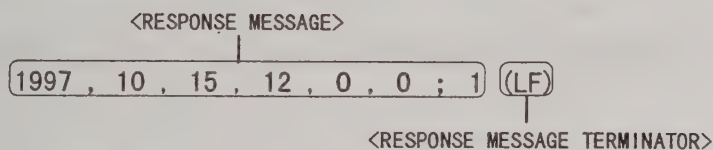
■ Separators

Two functional elements are available as separators as described below.



Functional element	Explanation
<RESPONSE MESSAGE UNIT SEPARATOR>	Separates multiple <RESPONSE MESSAGE UNIT> elements from one another. Represented by a semicolon. Example: NORM ; PRBS7
<RESPONSE DATA SEPARATOR>	Separates multiple <RESPONSE DATA> elements from one another. Represented by a comma.

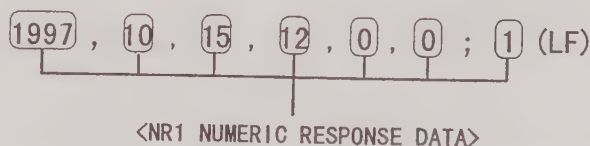
■ Terminator



Functional element	Explanation
<RESPONSE MESSAGE TERMINATOR>	Attached at the end of a <RESPONSE MESSAGE> to complete the message transfer. Represented by an LF (or LF + EOI for GPIB).

■ Response data

Response data supported by this instrument is grouped into six functional elements as described below. In detailed descriptions of commands, the IEEE488.2-prescribed response data types associated with device-specific parameter types are given.



Functional element	Explanation
<CHARACTER RESPONSE DATA>	A short character string associated with a parameter function.
<NR1 NUMERIC RESPONSE DATA>	Represents a decimal integer.
<NR2 NUMERIC RESPONSE DATA>	Represents a fixed-point decimal number.
<HEXADECIMAL NUMERIC RESPONSE DATA>	Represents a hexadecimal number, which is expressed by #H+{A through F, a through f, 0 through 9).
<STRING RESPONSE DATA>	Represents a character string enclosed in double quotation marks.
<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>	Represents 8-bit binary data with a known length. #YYYYYY<binary> X : Number of digits of YYYYYY YYYYYY : Byte length of <binary> <binary> : Binary data Examples: #14<DAB><DAB><DAB><DAB> <DAB> : Data byte
<ARBITRARY ASCII RESPONSE DATA>	Represents undelimited 7-bit ASCII text data.

Note

With SCPI instruments, the character strings represented by <CHARACTER PROGRAM DATA> and <STRING PROGRAM DATA> have long and short forms. A short form is an abbreviation of the corresponding long form.

If a program command has any one of the types of parameters defined above and a response to the corresponding query command is <CHARACTER RESPONSE DATA> or <STRING RESPONSE DATA>, the response to the query would be a character string in its short form.

4.4 Common Commands

The table below divides the IEEE488.2 common commands supported by this instrument into a number of functional groups.

Table 4-3 IEEE488.2 Common Command List

Group	Group function	Mnemonic
System data	Provides device-specific information on the device connected to the GPIB system, including the device manufacturer's name, model name, and serial number.	*IDN?
Internal operation	Internal device controls: level 3 device reset, and internal device self-testing and error detection	*RST *TST?
Synchronization	Device and controller synchronization by service request waiting, by device output queue response waiting, or by forced sequential execution.	*OPC *OPC? *WAI
Status and event	The status byte is composed of summary bits each representing a status summary message. These summary bits are set or reset from the standard event register, output queue, and extended event register or extended queue. Four commands and five queries are supported to set or clear or enable or disable data on these registers or queues and to determine the settings of these registers.	*CLS *ESE *ESE? *ESR? *SRE *SRE? *STB?
Device trigger	Defines the command sequence to be executed by the device when it receives the IEEE488.2 bus command GET.	*TRG
Option check	Queries the optional facilities installed in the device.	*OPT?

*IDN?

Response <ARBITRARY ASCII RESPONSE DATA>
 <product manufacture's name>,<model name>,<serial number>,<firmware
 revision number>
 This device returns ANKITSU,MP1630B,0,0001.

Function Reports the product manufacturer's name, model name, and so on. (This
 instrument responds with the mainframe model name.)

Example *IDN?

*RST

Parameter None

Function Resets the instrument settings to their factory defaults.

Example *RST

***TST?**

Response <NR1 NUMERIC RESPONSE DATA>
 0 : No error
 1 : Error occurrence

Function Queries the result of self-testing (such as a RAM check and a beep check) executed at power-on.

Example *TST?

***OPC**

Parameter None

Function Sets bit 0 (operation complete bit) of the standard event status register to turn ON SRQ when the execution of the preceding instruction is complete.

Example *OPC

***OPC?**

Response <NR1 NUMERIC RESPONSE DATA>
 1

Function Returns 1 when the execution of the preceding instruction is complete..

Example OPC?

***WAI**

Parameter None

Function Defers the start of execution of an instruction until the execution of the preceding instruction is complete.

Example *WAI

***CLS**

Parameter None

Function Clears all event registers and queues, except for the output queue and its MAV summary message. The device-specific status enable register and the transition filter are reset by the SCPI command :STATus:PRESet. If *CLS is transmitted right after <PROGRAM MESSAGE TERMINATOR> and before a <Query MESSAGE UNIT> element, both the output queue and the MAV bit are cleared.

Example *CLS

***ESE**

Parameter <DECIMAL NUMERIC PROGRAM DATA>

Integer from 0 to 255

The sum total of the values of the bits that are enabled among all the bits of the standard event enable register serves as a parameter. The value of a bit that is disabled is 0.

The register settings of this instrument are as follows:

Bit 7 ($2^7 = 128$)	Power-on
Bit 6 ($2^6 = 64$)	Local control execution(Not use for this instrument)
Bit 5 ($2^5 = 32$)	Command error
Bit 4 ($2^4 = 16$)	Execution error
Bit 3 ($2^3 = 8$)	Error other than command, query, and execution errors
Bit 2 ($2^2 = 4$)	Query error
Bit 0 ($2^0 = 1$)	Operation complete

Function Sets or clears the standard event status enable register.

Example Set bits 2 and 4 of the enable register.

*ESE 20

***ESE?**

Response <NR1 NUMERIC RESPONSE DATA>

0 to 255

Function Queries the current value of the standard event status enable register.

Example When *ESE? is transmitted after the execution of *ESE 20, 20 is returned.

***ESR?**

Response <NR1 NUMERIC RESPONSE DATA>

0 to 255

The sum total of the values of the bits of the standard status register serves as a response. See the parameter of the *ESE command for how to set bits of the standard status register.

Function Queries the current value of the standard event status register.

Example When a command error occurs, 32 is returned in response to the query of *ESE?.

***SRE**

Parameter <DECIMAL NUMERIC PROGRAM DATA>

Integer from 0 to 255

The sum total of the values of the bits that are enabled among all the bits of the service request enable register serves as a parameter. The value of a bit that is disabled is 0.

The register settings of this instrument are as follows:

- Bit7(2⁷=128) Operation status register summary
- Bit5(2⁵=32) Event status register summary
- Bit4(2⁴=16) Indicates the output queue is not empty
- Bit3(2³=8) Questionable status register summary
- Bit2(2²=4) Indicates the error/ event queue is not empty

Function Sets or clears the service request enable register.

Example Set bit 4 of the enable register.

*SRE 16

***SRE?**

Response <NR1 NUMERIC RESPONSE DATA>

0 to 255

Function Queries the current value of the service request enable register.

Example When *SRE? is transmitted after the execution of *SRE 16, 160 is returned.

***STB?**

Response <NR1 NUMERIC RESPONSE DATA>

Integer from 0 to 255

The sum total of the values of the bits that are enabled among all the bits of the status byte register serves as a response.

The register settings of this instrument are as follows:

- Bit7(2⁷=128) Operation status register summary
- Bit6(2⁶=64) MSS (Master Summary Status) summary message
- Bit5(2⁵=32) Event status register summary
- Bit4(2⁴=16) Indicates the output queue is not empty
- Bit3(2³=8) Questionable status register summary
- Bit2(2²=4) Indicates the error/event queue is not empty

Function Returns the current value of the status byte containing the MSS (Master Summary Status) bit.

Example If the event status register summary is true, 32 is returned in response to the query *STB?

***TRG**

Parameter None
Function Performs the same operation as does the IEEE488.1 bus command GET (Group Execute Trigger). For this instrument, starts measurement.
Example *TRG

***OPT?**

Response <ARBITRARY ASCII RESPONSE DATA>
Character associated with the option installed
0 No option
OPT01 GPIB option
OPT02 Ethernet option
Function Reports a list of all the options installed in the MP1630B mainframe, separated from each other by a comma.
Example If the GPIB interface option is installed, OPT01 is returned in response to the transmission of *OPT?

4.5 SCPI Summary

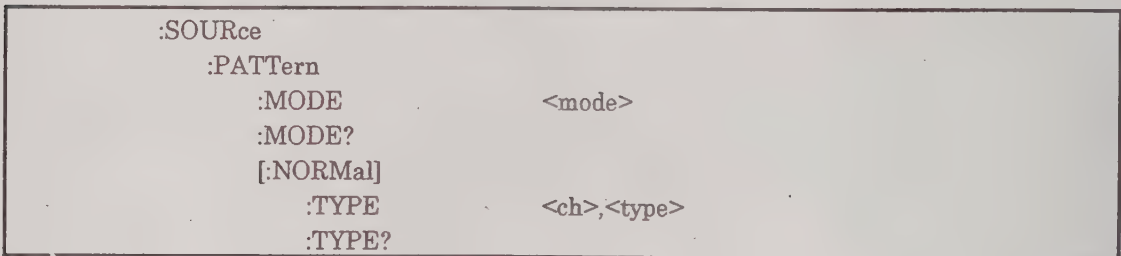
This instrument uses SCPI (Standard Commands for Programmable Instruments) to implement remote control.

This section gives a summarized description of SCPI, including its command notation.

Summary

SCPI is a hardware-independent device command language defined by the SCPI Consortium. Its objective is to speed up the development of programs for automatic testing equipment (ATE). To this end, it maintains consistency across the programming environment, including device control and data handling. It also provides compatibility among similar devices or among similarly functioning different devices, so that all similarly functioning devices can be controlled by implement an exactly the same sequence controls.

Command structure



The SCPI commands are based on a hierarchical structure, in which they are divided into functional groups called subsystems.

In this manual, each subsystem is represented by a command tree like the one shown above. Identical headers appearing in a tree each refer to a different function depending on where in the tree they appear. Accordingly, all commands must be expressed in a full-path format, including the header used.

Example:

SOURce subsystem

SOURce is the highest level node.

PATtern is the second highest level node.

MODE, MODE?, and NORMal are the third highest level nodes.

TYPE and TYPE? are the fourth highest level nodes.

<mode> designates the MODE parameter type, while <ch> and <type> represent the TYPE parameter types.

Command notation

:SOURce:PATtern:MODE <mode>	<mode>=NORMal,PON
:SOURce:PATtern:MODE	
:SOURce:PATtern[:NORMal]:TYPE <ch>,<type>	<ch> = 0 to 16
:SOURce:PATtern[:NORMal]:TYPE?	<type> = PRBS7,PRBS9,PRBS11,...

The commands shown above are involved in the command tree introduced earlier.

Rules of command notation are summarized below.

- **Command format**

A command begins with a colon (:). A command is composed of an interconnection of headers by a colon (:).

- **Header abbreviation**

Headers have both short and long forms. The short form of a header is an abbreviation of the corresponding long form covering only the uppercase letter portion of the header.

Whether short or long forms are used in a command, both variations of the command are interpreted to designate the same command. (The short and long forms of a header may be intermixed in a single command.)

Though distinctions are made between upper and lowercase letters in the command explanations, there are actually no such distinctions. (SOURCE, Source, and source are interpreted to designate the same header.)

Examples:

Long form	:SOURCE:PATtern:NORMAL:TYPE 1,PRBS11
Short form	:SOUR:PATT:NORM:TYPE 1,PRBS11
Long + short	:SOUR:PATtern:NORM:TYPE 1,PRBS11

- **Option node**

The brackets ([]) denote an option node. The header that is enclosed in brackets may be omitted. Whether option nodes are omitted from a command or not, both variations of the command are interpreted to designate the same command.

Examples:

When a header is not omitted	:SOURce:PATtern:NORMal:TYPE 1,PRBS11
When a header is omitted	:SOURce:PATtern:TYPE 1,PRBS11

- **Command separator**

At least one space is required between a command and the parameter that follows. Two or more parameters must be delimited from one another by a comma (,).

Compounded commands

Commands can be compounded by using a semicolon (;) as shown in the example below. The second command is referenced as being at the same level as the lowest level of the first command. Accordingly, while the second command can be expressed in a full-path format as in Example 1, it is also possible to leave the headers higher than MODE omitted as in Example 2.

Examples:

- 1) :SOURce:PATtern:MODE NORMal;;SOURce:PATtern:MODE?
- 2) :SOURce:PATtern:MODE NORMal;MODE?

Chapter 5 Status Reporting

5.1 MP1630B Status Register Configuration

The configuration of the status registers maintained by the instrument conforms to the specifications of SCPI. SCPI provides for the IEEE488.2-prescribed status register configuration and the availability of an SCPI-specific operational status register and a QUEStionable status register.

A simple schematic of the status registers maintained by the instrument is shown below.

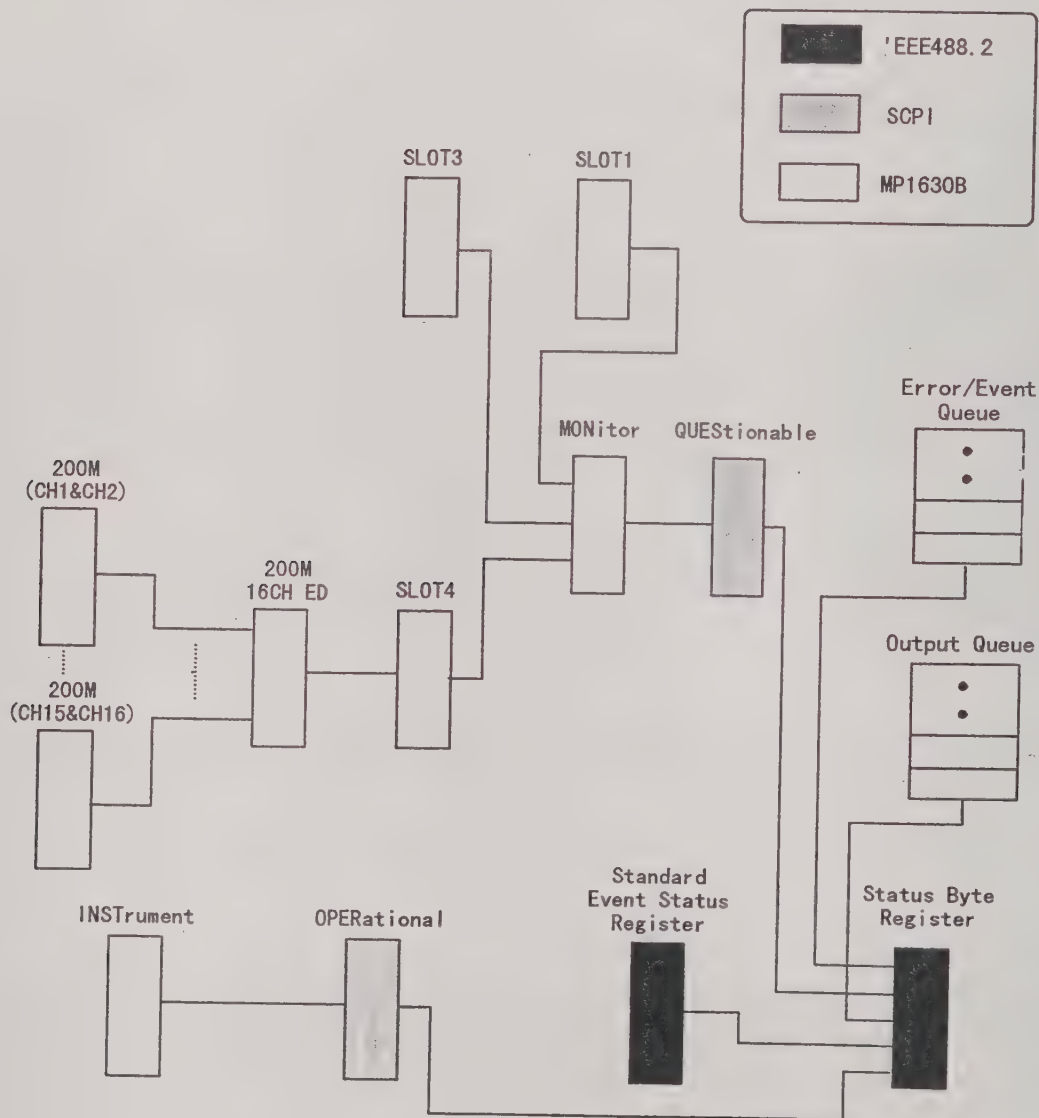


Figure 5-1 MP1630B Status Register Configuration

The configuration of the SCPI-prescribed and device-specific status registers is shown below.

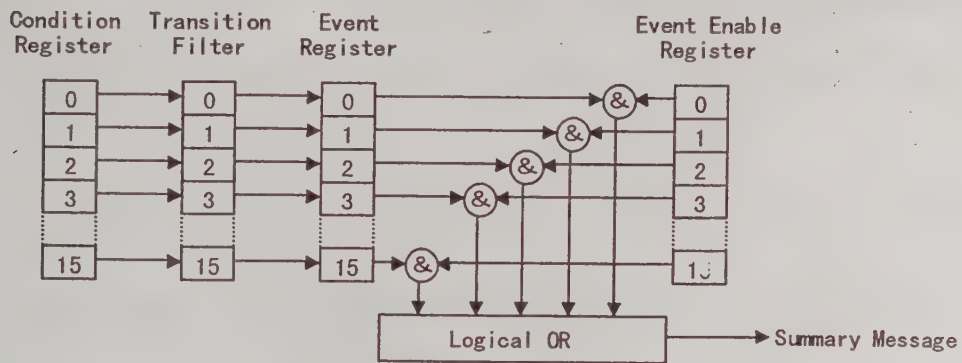


Figure 5-2 Status Register Configuration

Table 5-1 Register Filter Contents

Register filter	Description
Condition Register	Monitors the device status and changes on a real-time basis with changes in the device status. Therefore, this register does not store status information.
Transition Filter	Loads the event register with the contents of the condition register. The transition filter works in three ways depending on which changes in the condition register are tested. Positive changes : The event becomes true only if a false-to-true change occurs in the corresponding condition. Negative changes : The event becomes true only if a true-to-false change occurs in the corresponding condition. Two-way changes : The event becomes true if changes occur in either the positive or negative direction.
Event Register	Stores the output of the transition filter.
Event Enable Register	Specifies which bit of the corresponding event register should be set to make the corresponding summary message true.

5.2 IEEE488.2-prescribed Status Registers

IEEE488.2 prescribes the two registers described below.

Status byte register	Capable of setting RQS and seven summary message bits. Used in conjunction with the service request enable register so as to turn on SRQ when the result of their ORing is not zero. RQS is system-reserved by bit 6. The presence of a service request is reported to the external controller by way of this bit.
Standard event status register	Stores the eight kinds of standard events the device may encounter. logical OR output bits are summarized in bit 5 of the status byte register as an ESB (Event Status Bit) summary message.

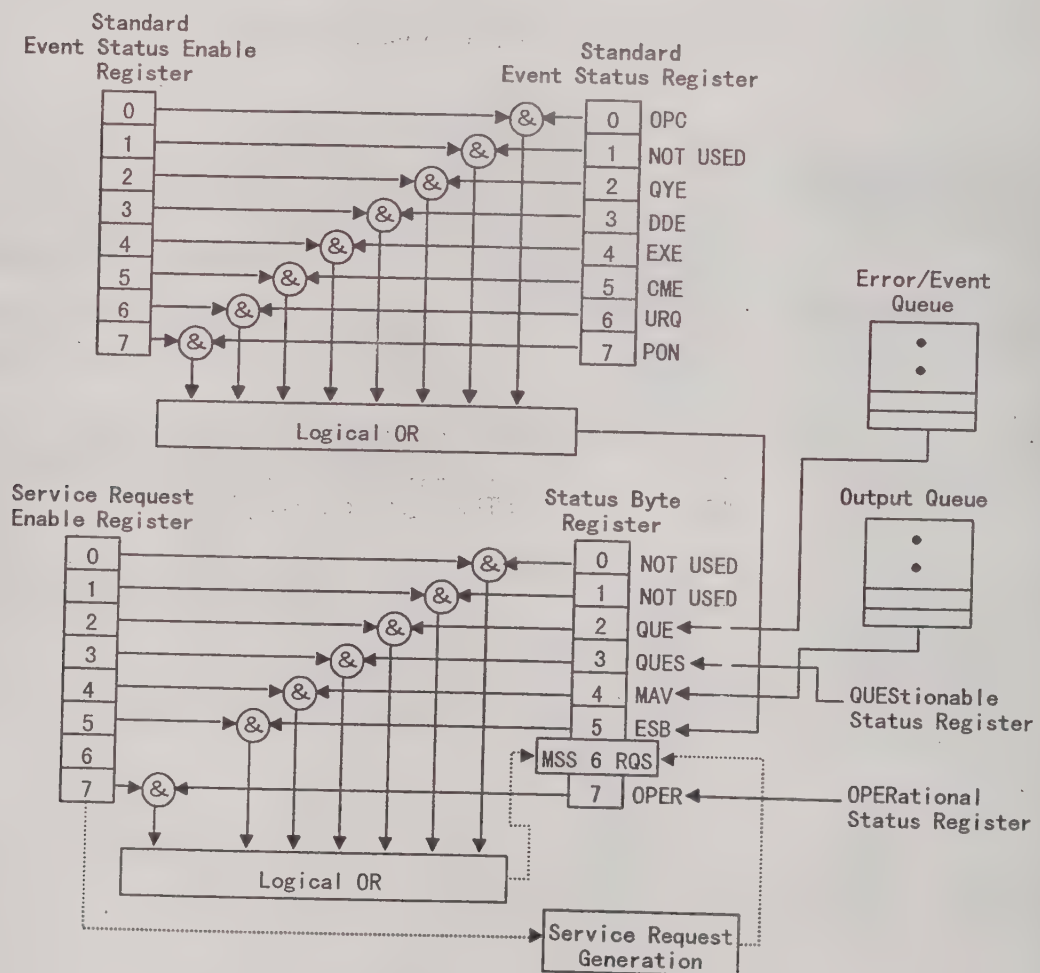


Figure 5-3 IEEE488.2-prescribed Status Registers

Table 5-2 Status Byte Register Bit Definitions

Bit	Mnemonic	Description
2	QUE (Error/Event QUEue)	Indicates the error/event queue is not empty.
3	QUES (QUESTionable status register summary)	QUESTionable status register summary.
4	MAV (Message AVailable)	Indicates the output queue is not empty. The MAV summary message bit is set to 1 (true) when the device becomes ready to receive a response message transmission request from the controller. The controller might transmit a query command to the device and then wait for MAV to become true.
5	ESB (Event Summary Bit)	Standard event status register summary
6	RQS (ReQuest Service) MSS (Master Summary Status)	RQS Returns a 7-bit status byte to the controller upon serial polling, along with this RQS message. MSS Indicates the device has at least one service request condition. This MSS appears in bit 6 of the response, in place of the RQS message, upon status byte query by the *STB? Command.
7	OPER (OPERation status register summary)	OPERation status register summary

Table 5-3 Standard Event Status Register Bit Definitions

Bit	Mnemonic	Description
0	OPC(Operation Complete)	Indicates the specified operation is complete.
2	QYE(Query Error)	Indicates a query error occurred.
3	DDE(Device-dependent Error)	Indicates an error other than command, query, and execution errors occurred.
4	EXE(Execution Error)	Indicates an execution error occurred.
5	CME(Command Error)	Indicates a command error has occurred.
6	URQ(User Request)	Indicates a local control was executed. Since this is not supported by this instrument, bit 6 is always set to 0.)
7	PON(Power on)	Indicates power was turned on.

5.3 SCPI-prescribed Status Registers

In addition to the status registers prescribed by IEEE488.2, SCPI provides for the availability of the registers listed below.

Table 5-4 SCPI-prescribed Status Registers

OPERation status register	Reports the device status, such as measuring. This register may be used to wait for the completion of any time-consuming operation, such as error/alarm measurement and eye margin measurement.
QUEStionable status register	Reports the status of a signal, such as one designating the result of a measurement. This register may be used to request a service from the controller when errors occur.

■ OPERation status register

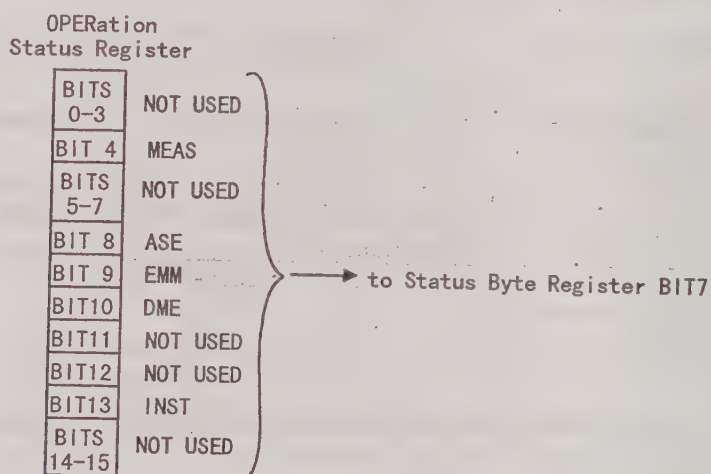


Figure 5-4 OPERation Status Register

Table 5-5 OPERation Status Register Bit Definitions

Bit	Mnemonic	Description
4	MEAS (MEASuring)	Indicates error/alarm measurement is in progress.
8	ASE (Auto SEaching)	Indicates an auto search is in progress.
9	EMM (Eye Margin Measuring)	Indicates eye margin measurement is in progress.
10	DME (Delay MEasuring)	Indicates delay measurement is in progress.
13	INST (INSTrument status register summary)	INSTrument status register summary

■ QUESTionable status register

QUESTionable
Status Register



Figure 5-5 QUESTionable Status Register

Table 5-6 QUESTionable Status Register Bit Definition

Bit	Mnemonic	Description
9	MON (MONitor status register summary)	MONitor status register summary

5.4 Device-specific Status Registers

As mentioned previously, device-specific registers are available, in addition to the IEEE488.2-prescribed and SCPI-prescribed status registers.

Table 5-7 Device-specific Status Registers

INSTRument status register	Reports the device status and measurement status.
MONitor status register	Reports the status of signals from the units mounted in the instrument.
SLOT1 status register	Reports the status of signals from the unit inserted in slot1. Clock generator alarms can be read.
SLOT3 status register	Reports the status of signals from the unit inserted in slot3. Alarms in the PPG inserted in slot3 can be read.
SLOT4 status register	Reports the status of signals from the unit inserted in slot4. Errors/alarms in the ED inserted in slot4 can be read. °
200M 16CH ED (slot4) status register	Reports the status of signals from the 200 16CH ED unit inserted in slot4. Errors/alarms in the 200M 16CH ED inserted in slot4 can be read.
200M CH1&CH2 to CH15&CH16 (slot4) status registers	Report the status the CH1 to CH16 signals from the 200M 16CH ED unit inserted in slot4. Errors/alarms in the individual channels in the 200M 16CH ED inserted in slot4 can be read.

■ INSTRUMENT status register

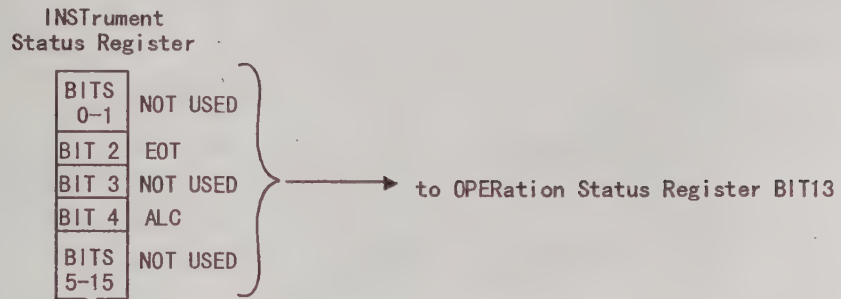


Figure 5-6 INSTRUMENT Status Register

Table 5-8 INSTRUMENT Status Register Bit Definitions

Bit	Mnemonic		Description
2	EOT	(End of Test period)	Marks the end of testing (measurement).
4	ALC	(Alarm Change)	Indicates the occurrence of an alarm change.

■ MONitor status register

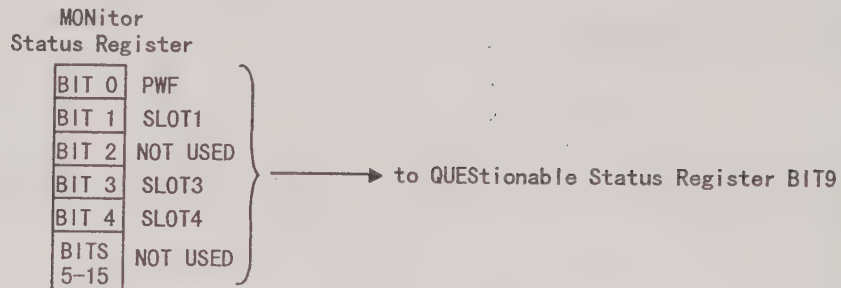


Figure 5-7 MONitor Status Register

Table 5-9 MONitor Status Register Bit Definitions

Bit	Mnemonic	Description
0	PWF (Power Fail)	Indicates a power failure occurred.
1	SLOT1	SLOT1 status register summary. Indicates some event occurred in the unit (clock generator) inserted in slot1.
3	SLOT3	SLOT3 status register summary. Indicates some event occurred in the unit (PPG) inserted in slot3.
4	SLOT4	SLOT4 status register summary. Indicates some event occurred in the unit (ED) inserted in slot4.

■ SLOT1 status register

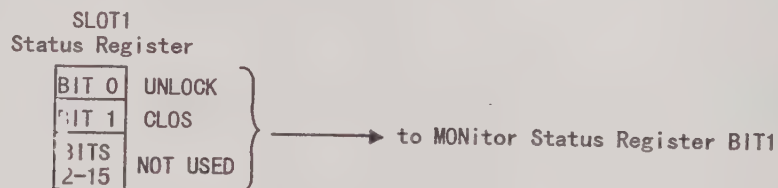


Figure 5-8 SLOT1 Status Register

Table 5-10 SLOT1 Status Register Bit Definitions

Bit	Mnemonic	Description
0	UNLOCK (PLL UNLOCK)	Indicates the occurrence of a PLL unlock.
1	CLOS (External Clock LOSs)	Indicates the occurrence of an external clock loss.

■ SLOT3 status register

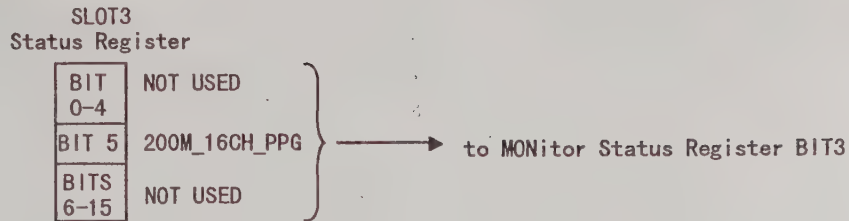


Figure 5-9 SLOT3 Status Register

Table 5-11 SLOT3 Status Register

Bit	Mnemonic	Description
5	200M_16CH_PPG	Indicates the occurrence of a clock loss in the 200M 16CH PPG.

■ SLOT4 status register

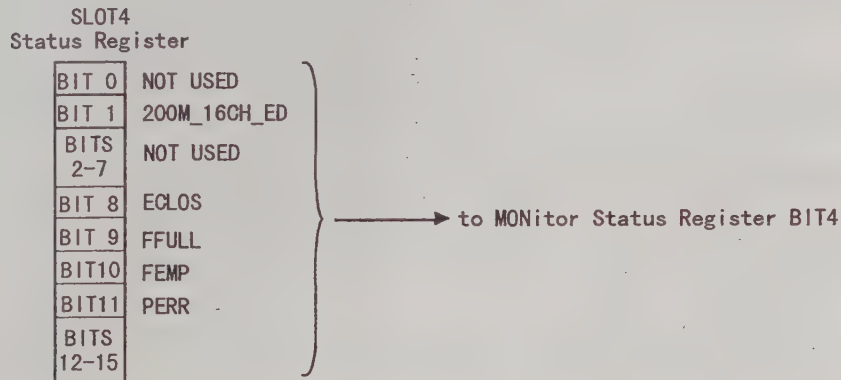


Figure 5-10 SLOT4 Status Register

Table 5-12 SLOT4 Status Register Bit Definitions

Bit	Mnemonic	Description
1	200M_16CH_ED	200M 16CH ED status register summary
8	ECLOS	Indicates the occurrence of a ED Clock Loss
9	FFULL	Indicates the occurrence of a FIFO Full
10	FEMP	Indicates the occurrence of a FIFO Empty
11	PERR	Indicates the occurrence of a Phase Error

■ 200M 16CH ED status register/

200M CH1&CH2 to CH15&CH16 status registers

Relationships between the 200M 16CH ED status register and the 200M CH1&CH2 to CH15&CH16 status registers are shown below.

200M CH1&CH2
Status Register

BIT 0	IERR_1
BIT 1	OERR_1
BIT 2	AERR_1
BIT 3	PRBS_1
BIT 4	PRGM_1
BIT 5	CLOS_1
BIT 6	IERR_2
BIT 7	OERR_2
BIT 8	AERR_2
BIT 9	PRBS_2
BIT10	PRGM_2
BIT11	CLOS_2
BITS 12-15	NOT USED

200M CH15&CH16
Status Register

BIT 0	IERR_15
BIT 1	OERR_15
BIT 2	AERR_15
BIT 3	PRBS_15
BIT 4	PRGM_15
BIT 5	CLOS_15
BIT 6	IERR_16
BIT 7	OERR_16
BIT 8	AERR_16
BIT 9	PRBS_16
BIT10	PRGM_16
BIT11	CLOS_16
BITS 12-15	NOT USED

BIT 0	CH1_2
BIT 1	CH3_4
BIT 2	CH5_6
BIT 3	CH7_8
BIT 4	CH9_10
BIT 5	CH11_12
BIT 6	CH13_14
BIT 7	CH15_16
BITS 8-15	NOT USED

to SLOT4
Status Register BIT1

Table 5-13 200M 16CH ED Status Register Bit Definitions

Bit	Mnemonic	Description
0	CH1_2	200M CH1&CH2 status register summary
1	CH3_4	200M CH3&CH4 status register summary
2	CH5_6	200M CH5&CH6 status register summary
3	CH7_8	200M CH7&CH8 status register summary
4	CH9_10	200M CH9&CH10 status register summary
5	CH11_12	200M CH11&CH12 status register summary
6	CH13_14	200M CH13&CH14 status register summary
7	CH15_16	200M CH15&CH16 status register summary

Table 5-14 200M CH1&CH2 to CH15&CH16 Status Register Bit Definitions

Bit	Mnemonic	Description
0	IERR_N	Indicates the occurrence of an INS error in CH(N).
1	OERR_N	Indicates the occurrence of an OMI error in CH(N).
2	AERR_N	Indicates the occurrence of an ALL error in CH(N).
3	PRBS_N	Indicates the occurrence of a pattern sync loss (PRBS) in CH(N).
4	PRGM_N	Indicates the occurrence of a pattern sync loss (PRGM) in CH(N).
5	CLOS_N	Indicates the occurrence of a clock loss in CH(N).
6	IERR_N+1	Indicates the occurrence of an INS error in CH(N+1).
7	OERR_N+1	Indicates the occurrence of an OMI error in CH(N+1).
8	AERR_N+1	Indicates the occurrence of an ALL error in CH(N+1).
9	PRBS_N+1	Indicates the occurrence of a pattern sync loss (PRBS) in CH(N+1).
10	PRGM_N+1	Indicates the occurrence of a pattern sync loss (PRGM) in CH(N+1).
11	CLOS_N+1	Indicates the occurrence of a clock loss in CH(N+1).

5.5 Reading and Writing Status Registers

The table below summarizes the methods of reading and writing the individual status registers.

Table 5-15 Reading and Writing Status Registers

Register	How to read	How to write
Status byte register	Use serial polling to read. A 7-bit status byte is returned to the controller upon serial polling, along with the RQS message bit. The status byte value does not change at this time. *STB? common query The contents of the status byte register are returned, along with a single numeric value consisting of an MSS summary message.	Not permitted
Service request enable register	*SRE? common query The register contents do not change at this time.	*SRE common command
Standard event status register	*ESR? common query The register is cleared after it is read.	Not permitted
Standard event status enable register	*ESE? common query The register contents do not change at this time.	*ESE common command
SCPI event register	Device-specific command :STATus:...:EVENT? The register is cleared at this time.	Not permitted
SCPI condition register	Device-specific command :STATus:...:CONDition?	Not permitted
SCPI enable register	Device-specific command :STATus:...:ENABLE? The register is cleared at this time.	Device-specific command :STATus:...:ENABle
SCPI transition filter	Device-specific commands :STATus:...:PTRansition? :STATus:...:NTRansition? The register is cleared at this time.	Device-specific commands :STATus:...:PTRansition :STATus:...:NTRansition
Error/event queue	Device-specific command :SYSTem:ERRor?	Not permitted

Note :

- The SCPI... registers mentioned above include SCPI-prescribed and device-specific status registers.
- See Chapter 4 for more details on common commands and queries and see Chapter 6 for more details on device-specific commands.

5.6 Clearing and Resetting Status Registers

The table below summarizes the methods of clearing and resetting the individual status registers, and the scopes of influences of such clearing and resetting.

Table 5-16 Clearing and Resetting Status Registers

Register	*RST	*CLS	Power-on	STATUS:PRESet	Other way to clear
Status byte register	No change	Clear	Clear	No change	
Service request enable register	No change	No change	Clear	No change	Execute *SRE 0.
Standard event status register	No change	Clear	Clear	No change	Cleared when the event is read with *ESR?.
Standard event status enable register	No change	No change	Clear	No change	Execute *ESE 0.
SCPI event register	No change	Clear	Clear	No change	Cleared when the event is read with :STATUS:...EVENT?.
SCPI enable register	No change	No change	Reset	Reset	Execute :STATUS:...ENABLE 0.
SCPI transition filter	No change	No change	Reset	Reset	: Execute :STATUS:...PTRansition 0 and :STATUS:...NTRansition 0.
Error/event queue	No change	Clear	Clear	No change	Read all events with :SYSTem:ERRor?

Note:

- The SCPI... registers mentioned above include SCPI-prescribed and device-specific status registers.

The table lists the values to which the registers are reset by the :STATUS:PRESet command.

Table 5-17 Values to which Registers Are Reset by :STATUS:PRESet Command

Register	Enable/filter	Reset value
OPERational status register	Enable register	All 0
QUESTionable status register	PTRansition filter	All 1
	NTRansition filter	All 0
INSTrument status register	Enable register	ALL 1
	PTRansition filter	ALL 1
	NTRansition filter	ALL 0
Other status registers	Enable register	All 1
	PTRansition filter	All 1
	NTRansition filter	All 1

Chapter 5 Status Reporting

Table 5-1: Status Reporting Table

Table 5-2: Status Reporting Table

Table 5-3: Status Reporting Table

Table 5-4: Status Reporting Table

Table 5-5: Status Reporting Table

Table 5-6: Status Reporting Table

Table 5-7: Status Reporting Table

Table 5-8: Status Reporting Table

Table 5-9: Status Reporting Table

Table 5-10: Status Reporting Table

Table 5-11: Status Reporting Table

Table 5-12: Status Reporting Table

Table 5-13: Status Reporting Table

Table 5-14: Status Reporting Table

Table 5-15: Status Reporting Table

Table 5-16: Status Reporting Table

Table 5-17: Status Reporting Table

Table 5-18: Status Reporting Table

Table 5-19: Status Reporting Table

Table 5-20: Status Reporting Table

Chapter 6 Device Message Details

6.1 Command List

6.1.1 Subsystem Configuration

The device-specify commands provided by the instrument conform to the specifications of SCPI.

The SCPI commands are divided into functional groups called subsystems, each encompassing a tree of commands to provide detailed functions.

This instrument supports both subsystems that provide unit-specific setups/queries and those that provide unit-common setups/queries as listed in the table below.

Table 6-1 Subsystem Configuration

	200M 16CH PPG	200M 16CH ED	200M CG
Logical signal setup (including pattern setup)	SOURce	SENSe	
Electrical signal setup (including clock I/F setup)	OUTPut	INPut OUTPut	OUTPut
Measurement result reading, etc.	CALCulate	CALCulate	CALCulate
Arming setup			ARM
Common setup	INSTRument	INSTRument	
Display handling	DISPlay		
Other setup (including printout and date)	SYSTem		
Status register setup	STATus		

6.1.2 Command Trees

The instrument's command trees are listed in the following order, beginning with the next page:

1. 200M 16CH PPG (MU163020B) unit setup/query commands
2. 200M 16CH ED (MU163040B) unit setup/query commands
3. 200M clock generator (MU163020A) unit setup/query commands
4. Other setup/query commands

200M 16CH PPG (ML163020B) Unit Setup/Query Command Tree

■ SOURCE subsystem

KEYWORD	PARAMETER FORM
:SOURCE	
:PATtern	
:MODE	<mode>
:MODE?	
[:NORMal]	
:OMODE	<mode>
:OMODE?	
:TYPE	<ch>,<type>
:TYPE?	<ch>
:PRBS	
:MRATio	<ch>,<mratio>
:MRATio?	<ch>
:BSHift	<ch>,<numeric>
:BSHift?	<ch>
:ZSUBstitute	
:LENGth	<ch>,<numeric>
:LENGth?	<ch>
:ZLENGth	<ch>,<numeric>
:ZLENGth?	<ch>
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:PROGram	
:LENGth	<ch>,<numeric>
:LENGth?	<ch>
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:MIX	
:PROGram	
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:PRBS	
:TYPE	<ch>,<type>
:TYPE?	<ch>
:MRATio	<ch>,<mratio>
:MRATio?	<ch>
:BSHift	<ch>,<numeric>
:BSHift?	<ch>
:BLOCK	
:SIZE	<ch>,<row>,<column>
:SIZE?	<ch>
:SET	<ch>,<column>,<patt>
:SET?	<ch>,<column>
:LENGth	<ch>,<column>,<numeric>
:LENGth?	<ch>,<column>

KEYWORD	PARAMETER FORM
:SOURce	
:PATtern	
[:NORMal]	
:BURSt	
:MODE	<mode>
:MODE?	
:CYCLe	<numeric>
:CYCLe?	
:ELENgth	<numeric>
:ELENgth?	
:PMODE	<mode>
:PMODE?	
:PDIFference	
:PHASe	<channel>,<phase>
:PHASe?	<channel>
:PON	
:ECHannel	<numeric>
:ECHannel?	
:PLENgth	<ch>,<numeric>
:PLENgth?	<ch>
:PROGram	
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:PRBS	
:TYPE	<ch>,<type>
:TYPE?	<ch>
:MRATio	<ch>,<ratio>
:MRATio?	<ch>
:BSHift	<ch>,<numeric>
:BSHift?	<ch>
:BLOCK	
:SIZE	<ch>,<numeric>
:SIZE?	<ch>
:SET	<ch>,<blk>,<patt>
:SET?	<ch>,<blk>
:LENgth	<ch>,<blk>,<numeric>
:LENgth?	<ch>,<blk>

KEYWORD**PARAMETER FORM**

:SOURce	
:PATtern	
:PON	
:OTIMing	
:BMODE	<mode>
:BMODE?	
:BCYCLE	<numeric>
:BCYCLE?	
:GTIMe	<gt>,<numeric>
:GTIMe?	<gt>
:DATA	
:CHANnel	<ch>,<start>,<end>,<data>
:CHANnel?	<ch>,<start>
:BLOCK	<ch>,<blk>,<start>,<end>,<data>
:BLOCK?	<ch>,<blk>,<start>
:BDATa	
:CHANnel	<ch>,<start>,<end>,<bdata>
:CHANnel?	<ch>,<start>
:BLOCK	<ch>,<blk>,<start>,<end>,<bdata>
:BLOCK?	<ch>,<blk>,<start>
:DREVerse	
:CHANnel	<ch>
:BLOCK	<ch>,<blk>
:ADDReSS	<ch>,<start>,<end>
:DELTA	<ch>,<start>,<delta>
:ADDReSS	
:CHANnel?	<ch>
:BLOCK?	<ch>,<blk>
:DELay	
:CYCLE	<numeric>
:CYCLE?	
:TYPE	<type>
:TYPE?	

KEYWORD

:SOURce

:PATTern

:EADDITION

:SET

:SET?

:SINGLE

:MODE1

:MODE1?

:MODE2

:MODE2?

:MODE3

:MODE3?

:RATE

:RATE?

:BCYCLE

:BCYCLE?

:ELENGTH

:ELENGTH?

:ADDITION

:ADDITION?

:ABLOCK

:ABLOCK?

:BLOCK

:BLOCK?

PARAMETER FORM

<boolean>

<mode1>

<mode2>

<mode3>

<rate>

<numeric>

<numeric>

<ch>, <boolean>

<ch>

<ch>, <mode>

<ch>

<ch>, <blk>, <boolean>

<ch>

■ **OUTPUT subsystem****KEYWORD**

:OUTPut

:SET

:SET?

:CLOCK

:POLarity

:POLarity?

:OMODE

:OMODE?

:OUTPut

:OUTPut?

:DELAY

:DELAY?

:LEVEL

:LEVEL?

:TERMination

:TERMination?

:AMPLitude

:AMPLitude?

:OFFSet

:OFFSet?

:OREFERENCE

:OREFERENCE?

PARAMETER FORM

<boolean>

<ch>, <pol>

<ch>

<ch>, <mode>

<ch>

<ch>, <boolean>

<ch>

<ch>, <numeric>

<ch>

<ch>, <level>

<ch>

<ch>, <term>

<ch>

<ch>, <numeric>

<ch>

<ch>, <numeric>

<ch>

<ch>, <offset>

<ch>

KEYWORD	PARAMETER FORM
:OUTPut	
:DATA	
:MODE	<mode>
:MODE?	
:OUTPut	<ch>,<boolean>
:OUTPut?	<ch>
:SKEW	<ch>,<numeric>
:SKEW?	<ch>
:LEVel	<ch>,<level>
:LEVel?	<ch>
:TERMination	<ch>,<term>
:TERMination?	<ch>
:AMPLitude	<ch>,<numeric>
:AMPLitude?	<ch>
:OFFSet	<ch>,<numeric>
:OFFSet?	<ch>
:OREFERENCE	<ch>,<offset>
:OREFERENCE?	<ch>
:AUXiliary	
:LEVel	<level>
:LEVel?	
:CHANnel	<port>,<ch>
:CHANnel?	<port>
:TYPE	<port>,<source>
:TYPE?	<port>
:LOGic	<port>,<logic>
:LOGic?	<port>
:BLOCation	<port>,<numeric>
:BLOCation?	<port>
:ELOCation	<port>,<numeric>
:ELOCation?	<port>
:ENVelop	
:PDELay	<port>,<numeric>
:PDELay?	<port>
:NDELay	<port>,<numeric>
:NDELay?	<port>
:RESet	
:PWIDth	<port>,<numeric>
:PWIDth?	<port>
:LOCation	<port>,<loc>
:LOCation?	<port>

KEYWORD

:OUTPut

:SYNc

:SOURce

:SOURce?

:PATtern

:CHANnel

:CHANnel?

:TYPE

:TYPE?

:TRIGger

:SOURce

:SOURce?

:CHANnel

:CHANnel?

:BLOCK

:BLOCK?

:BTRigger

:DElay

:DElay?

PARAMETER FORM

<source>

<ch>

<type>

<source>

<ch>

<blk>

<ch>,<numeric>

<ch>

■ **CALCulate subsystem****KEYWORD**

:CALCulate

:DATA

:MONitor?

PARAMETER FORM■ **INSTrument subsystem****KEYWORD**

:INSTrument

:COUPle

:PATtern

:SET

:SET?

PARAMETER FORM

<boolean>

200M 16CH ED (MU163040B) Unit Setup/Query Command Tree

■ SENSE subsystem

KEYWORD	PARAMETER FORM
:SENSe	
:PATtern	
:MODE	<mode>
:MODE?	
[:NORMal]	
:IMODE	<mode>
:IMODE?	
:TYPE	<ch>,<type>
:TYPE?	<ch>
:PRBS	
:MRATio	<ch>,<mratio>
:MRATio?	<ch>
:BShifT	<ch>,<numeric>
:BShifT?	<ch>
:ZSUBstitute	
:LENGth	<ch>,<numeric>
:LENGth?	<ch>
:ZLENGth	<ch>,<numeric>
:ZLENGth?	<ch>
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:PROGram	
:LENGth	<ch>,<numeric>
:LENGth?	<ch>
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:MIX	
:PROGram	
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:PRBS	
:TYPE	<ch>,<type>
:TYPE?	<ch>
:MRATio	<ch>,<mratio>
:MRATio?	<ch>
:BShifT	<ch>,<numeric>
:BShifT?	<ch>
:BLOCK	
:SIZE	<ch>,<row>,<column>
:SIZE?	<ch>
:SET	<ch>,<column>,<patt>
:SET?	<ch>,<column>
:LENGth	<ch>,<column>,<numeric>
:LENGth?	<ch>,<column>

KEYWORD	PARAMETER FORM
:SENSe	
:PATtern	
[:NORMal]	
:SYNC	
:ASYNc	
:PROGram	<ch>,<boolean>
:PROGram?	<ch>
:PRBS	<ch>,<boolean>
:PRBS?	<ch>
:THReshold	
:PROGram	<ch>,<thre>
:PROGram?	<ch>
:PRBS	<ch>,<thre>
:PRBS?	<ch>
:PSMode	<ch>,<mode>
:PSMode?	<ch>
:BSMode	<ch>,<mode>
:BSMode?	<ch>
:FLENgth	<ch>,<numeric>
:FLENgth?	<ch>
:BURSt	
:ELENgth	<numeric>
:ELENgth?	
:PMODE	<mode>
:PMODE?	
:PON	
:ECHannel	<numeric>
:ECHannel?	
:PROGram	
:LOGic	<ch>,<logic>
:LOGic?	<ch>
:PRBS	
:TYPE	<ch>,<type>
:TYPE?	<ch>
:MRATio	<ch>,<ratio>
:MRATio?	<ch>
:BSHift	<ch>,<numeric>
:BSHift?	<ch>

KEYWORD

PARAMETER FORM

:SENSe	
:PATtern	
:PON	
:BLOCK	
:SIZE	<ch>,<numeric>
:SIZE?	<ch>
:SET	<ch>,<blk>,<patt>
:SET?	<ch>,<blk>
:LENGth	<ch>,<blk>,<numeric>
:LENGth?	<ch>
:SYNC	
:ASYNc	
:PROGram	<ch>,<boolean>
:PROGram?	<ch>
:PRBS	<ch>,<boolean>
:PRBS?	<ch>
:THReshold	
:PROGram	<ch>,<thre>
:PROGram?	<ch>
:PRBS	<ch>,<thre>
:PRBS?	<ch>
:BSMode	<ch>,<mode>
:BSMode?	<ch>
:FLENGth	<ch>,<numeric>
:FLENGth?	<ch>
:DATA	
:CHANnel	<ch>,<start>,<end>,<data>
:CHANnel?	<ch>,<start>
:BLOCK	<ch>,<blk>,<start>,<end>,<data>
:BLOCK?	<ch>,<blk>,<start>
:BDATa	
:CHANnel	<ch>,<start>,<end>,<bdata>
:CHANnel?	<ch>,<start>
:BLOCK	<ch>,<blk>,<start>,<end>,<bdata>
:BLOCK?	<ch>,<blk>,<start>
:DREVerse	
:CHANnel	<ch>
:BLOCK	<ch>,<blk>
:ADDRes	<ch>,<start>,<end>
:DELTA	<ch>,<start>,<delta>
:ADDRes	
:CHANnel?	<ch>
:BLOCK?	<ch>,<blk>

KEYWORD	PARAMETER FORM
:SENSe	
:MEASure	
:TEST	<item>
:TEST?	
:START	
:STOP	
:EALarm	
:MODE	<mode>
:MODE?	
:PERiod	<d>,<h>,<m>,<s>
:PERiod?	
:BTIMe	
:SET	<boolean>
:SET?	
:START	<y>,<m>,<d>,<h>,<m>,<s>
:START?	
:ERRor	
:MODE	<ch>,<mode>
:MODE?	<ch>
:DBLock	<ch>,<mode>
:DBLock?	<ch>
:BLOCK	<ch>,<blk>,<boolean>
:BLOCK?	<ch>
:START?	
:STOP?	
:STATe?	
:ELAPsed?	
:TIMed?	
:ITIMe?	
:DELay1	
:MODE	<mode>
:MODE?	
:TOUT	<numeric>
:TOUT?	
:CHANnel	<ch>,<boolean>
:CHANnel?	
:STATe?	
:MSTATe?	<ch>
:START?	
:STOP?	
:ELAPsed?	
:TYPE	<type>
:TYPE?	

KEYWORD	PARAMETER FORM
:SENSe	
:MEASure	
:DELay2	
:MODE	<mode>
:MODE?	
:TOUT	<numeric>
:TOUT?	
:PATtern	<patt>
:PATtern?	
:CHANnel	<ch>
:CHANnel?	
:STATe?	
:MSTate?	
:START?	
:STOP?	
:ELAPsed?	
:TYPE	<type>
:TYPE?	
:EMARgin	
:MODE	<mode>
:MODE?	
:TYPE	<type>
:TYPE?	
:MARGin	
:THReshold	<thre>
:THReshold?	
:RESolution	<type>
:RESolution?	
:DIAGram	
:THReshold	<thre>,<boolean>
:THReshold?	
:POINT?	<thre>
:UNIT	<unit>
:UNIT?	
:CHANnel	<ch>
:CHANnel?	
:STATe?	
:START?	
:STOP?	
:ELAPsed?	

KEYWORD**PARAMETER FORM**

:SENSe

:MEASure

:ASEarch

:MODE

<mode>

:MODE?

:CHANnel

<channel>,<boolean>

:CHANnel?

:STARt

:STOP

:PATTern

<boolean>

:PATTern?

:STATe?

:FCHannel?

:RDIagram

:CHANnel?

:THReshold?

:POINT?

<thre>

:STARt?

:STOP?

:ANALyze

:CHANnel

<channel>,<boolean>

:CHANnel?

:TTEXT

:DATA

<data>,<boolean>

:DATA?

:IDATa

:ITEM

<item>,<boolean>

:ITEM?

:PERiod

<numeric>,<suffix>

:PERiod?

:EDATa

:ITEM

<item>,<boolean>

:ITEM?

:SDATa

:THReshol d

<thre>

:THReshold?

■ INPut subsystem

KEYWORD

PARAMETER FORM

:INPut

:CLOCK

:MODE

<mode>

:MODE?

:IMODE

<ch>,<mode>

:IMODE?

<ch>

:POLarity

<ch>,<pol>

:POLarity?

<ch>

:DELaY

<ch>,<numeric>

:DELaY?

<ch>

:TERMination

<ch>,<term>

:TERMination?

<ch>

:DATA

:MODE

<mode>

:MODE?

:LEVel

<ch>,<level>

:LEVel?

<ch>

:THReshold

<ch>,<numeric>

:THReshold?

<ch>

:TERMination

<ch>,<term>

:TERMination?

<ch>

:RCLOCK

:SOURCE

<source>

:SOURCE?

■ CALCulate subsystem

KEYWORD

:CALCulate

:EALarm

:SLEValuation

:SLEValuation?

:CLEValuation

:CLEValuation?

:ERRor

:INTerval

:INTerval?

:PERFormance

:MEASurement

:MEASurement?

:THReshold

:THReshold?

:TEINTerval

:MEASurement

:MEASurement?

:PFAil

:PFAil?

:DATA

:EALarm?

:EMARgin?

:DELay1?

:DELay2?

:START?

:SIZE?

:TEXT?

:MONitor?

:STORe

:CLEar

PARAMETER FORM

<boolean>

<boolean>

<numeric>,<suffix>

<boolean>

<string>

<boolean>

<boolean>

<channel>,<string>

<string>

<ch>

<type>

<type>,<channel>,<y>,<m>,<d>,<h>,<n>,<s>

<type>,<channel>,<y>,<m>,<d>,<h>,<n>,<s>

<ch>,<string>

<type>

<type>

■ OUTPut subsystem

KEYWORD	PARAMETER FORM
:OUTPut	
:SYNC	
:SOURce	<source>
:SOURce?	
:PATtern	
:CHANnel	<ch>
:CHANnel?	
:TYPE	<type>
:TYPE?	
:CLOCK	
:CHANnel	<ch>
:CHANnel?	

■ INSTrument subsystem

KEYWORD	PARAMETER FORM
:INSTrument	
:COUPle	
:PATtern	
:SET	<boolean>
:SET?	

200M Clock Generator (MU163020A) Unit Setup/Query Command Tree

■ OUTPut subsystem

KEYWORD

PARAMETER FORM

:OUTPut

:CLOCK

:SOURce

<source>

:SOURce?

:FREQuency

<numeric>

:FREQuency?

:RCLock

:SElect

<clock>

:SElect?

:JITTer

:EMODulation

<boolean>

:EMODulation?

:RANGe

<range>

:RANGe?

■ ARM subsystem

KEYWORD

PARAMETER FORM

:ARM

:SOURce

<source>

:SOURce?

:SET

<boolean>

:SET?

:STARt

■ CALCulate subsystem

KEYWORD

PARAMETER FORM

:CALCulate

:DATA

:MONitor?

<string>

Other Setup/Query Command Tree

■ DISPlay subsystem

KEYWORD	PARAMETER FORM
:DISPlay	
:ACHannel	<boolean>
:ACHannel?	
:WINDow	
:OPEN	<disp>
:OPEN?	
:CLOSe	<disp>
:SYSTem	
[:NAME]	<name>
[:NAME]?	
:SETup	
[:NAME]	<name>
[:NAME]?	
:CLOCK	
:CHANnel	<unit>,<ch>
:CHANnel?	<unit>
:DATA	
:CHANnel	<unit>,<ch>
:CHANnel?	<unit>
:PATTern	
:PAGE	<unit>,<page>
:PAGE?	<unit>
:CHANnel	<unit>,<ch>
:CHANnel?	<unit>
:AUXiliary	
:PORT	<unit>,<port>
:PORT?	<unit>
:TEST	
[:NAME]	<name>
[:NAME]?	
:EALarm	
:CHANnel	<unit>,<ch>
:CHANnel?	<unit>
:EADDition	
:CHANnel	<unit>,<ch>
:CHANnel?	<unit>

KEYWORD	PARAMETER FORM
:DISPlay	
:REStult	
:TIME	<type>
:TIME?	
:EALarm	
[:NAME]	<name>
[:NAME]?	
:MODE	<mode>
:MODE?	
:TYPE	<type>
:TYPE?	
:PITem	<pitem>
:PITem?	
:TITem	<titem>
:TITem?	
:UNIT	<unit>
:UNIT?	
:CHANnel	<channel>
:CHANnel?	
:ZOOM1 :ZOOM2 :ZOOM3 :ZOOM4	
:SET	<boolean>
:SET?	
:UNIT	<unit>
:UNIT?	
:CHANnel	<channel>
:CHANnel?	
:ITEM	<item>
:ITEM?	
:PITem	<pitem>
:PITem?	
:TITem	<titem>
:TITem?	
:HRESet	
:EMARgin	
:MODE	<mode>
:MODE?	
:ERATe	<erate>,<boolean>
:ERATe?	
:SCALE	
:ASCaLe	
:VOLTage	<min>,<step>
:VOLTage?	
:PHASe	<min>,<step>
:PHASe?	

KEYWORD	PARAMETER FORM
:DISPlay	
:RESult	
:EMARgin	
:MARKer	
:SET	<marker>,<boolean>
:SET?	<marker>
:MODE	<mode>
:MODE?	
:ERATe	<erate>
:ERATe?	
:MOVE	<marker>,<mode>
:POSition?	<marker>
:CUSTomize	
:SETup1 :SETup2 :SETup3 :SETup4 :SETup5 :SETup6	
:UNIT	<unit>
:UNIT?	
:CHANnel	<ch>
:CHANnel?	
:ITEM	<string>
:ITEM?	
:PATtern	
:UNIT	<unit>
:UNIT?	
:CHANnel	<ch>
:CHANnel?	
:OFFSet	<numeric>
:OFFSet?	
:BLOCK	<numeric>
:BLOCK?	
:RESult	
:UNIT	<unit>
:UNIT?	
:CH _Nnel	<channel>
:CH _Nnel?	
:ITEM	<item>
:ITEM?	
:ANALyze	
:MODE	<mode>
:MODE?	
:UNIT	<unit>
:UNIT?	
:CHANnel	<channel>
:CHANnel?	

KEYWORD

:DISPlay
 :ANALyze
 :TEXT
 :SCRoll
 :TIME
 :TIME?

PARAMETER FORM

<scroll>
 <year>, <month>, <day>, <hour>, <min>, <sec>

■ **SYSTem subsystem****KEYWORD**

:SYSTem
 :BEEPer
 :ERRor
 :SET
 :SET?
 :CHANnel
 :CHANnel?
 :ALARm
 :SET
 :SET?
 :CHANnel
 :CHANnel
 :SYSTem
 :SET
 :SET?
 :TYPE
 :TYPE?

PARAMETER FORM

<boolean>
 <unit>, <ch>, <boolean>
 <unit>
 <boolean>
 <unit>, <ch>, <boolean>
 <unit>
 <boolean>
 <type>, <boolean>

KEYWORD**PARAMETER FORM**

:SYSTem

:MODE?

:DATE

:DATE?

:TIME

:TIME?

:BSIZe?

:ERRor?

:VERSion?

:ORGanization

:HARDware?

:SOFTware?

:MMEMory

:RECall

:STORe

:QRECall

:QSTore

:CATalog?

:MEMory

:INITialize

:PRINt

:COPY

:TEXT

:EALarm

:TTEXT

:TIME

:ALL

:RTEXT

:TIME

:ALL

<year>,<month>,<day>

<hour>,<min>,<sec>

<mode>

<type>,<file_name>,<unit>,<ch>

<type>,<file_name>,<unit>

<file_name>

<file_name>,<comment>

<drv_dir>,<type>

<string>

<item>

<unit>,<channel>

<y1>,<m1>,<d1>,<h1>,<n1>,<s1>,<

<y2>,<m2>,<d2>,<h2>,<n2>,<s2>

<uni>,<channel>

<unit>,<channel>

<y1>,<m1>,<d1>,<h1>,<n1>,<s1>,<

<y2>,<m2>,<d2>,<h2>,<n2>,<s2>

<uni>,<channel>

KEYWORD**PARAMETER FORM**

:SYSTem

:PRINt

:PATTeRn

:TIME

:ADDReSS

<unit>,<start>,<end>

:ALL

<unit>

:STATe

:ADDReSS

<unit>,<start>,<end>

:ALL

<unit>

:DUMP

:ADDReSS

<unit>,<ch>,<form>,<start>,<end>

:ALL

<unit>,<ch>,<form>

:EMARgin

:REMargin

:DELaY1

:DELaY2

■ STATUS subsystem

KEYWORD

PARAMETER FORM

:STATus

:PRESet

:OPERation

[:EVENT]?

:CONDtion?

:ENABle

<numeric>

:ENABle?

:PTRansition

<numeric>

:PTRansition?

:NTRansition

<numeric>

:NTRansition?

:INSTrument

[:EVENT]?

:CONDtion?

:ENABle

<numeric>

:ENABle?

:PTRansition

<numeric>

:PTRansition?

:NTRansition

<numeric>

:NTRansition?

:QUESTionable

[:EVENT]?

:CONDtion?

:ENABle

<numeric>

:ENABle?

:PTRansition

<numeric>

:PTRansition?

:NTRansition

<numeric>

:NTRansition?

:MONitor

[:EVENT]?

:CONDtion?

:ENABle

<numeric>

:ENABle?

:PTRansition

<numeric>

:PTRansition?

:NTRansition

<numeric>

:NTRansition?

KEYWORD	PARAMETER FORM
:STATus	
:QUEStionable	
:MONitor	
:SLOT1	
[:EVENT]?	
:CONDtion?	
:ENABle	<numeric>
:ENABle?	
:PTRansition	<numeric>
:PTRansition?	
:NTRansition	<numeric>
:NTRansition?	
:SLOT3 :SLOT4	
[:EVENT]?	
:CONDtion?	
:ENABle	<numeric>
:ENABle?	
:PTRansition	<numeric>
:PTRansition?	
:NTRansition	<numeric>
:NTRansition?	
:M2CH16	
[:EVENT]?	
:CONDtion?	
:ENABle	<numeric>
:ENABle?	
:PTRansition	<numeric>
:PTRansition?	
:NTRansition	<numeric>
:NTRansition?	
:CH1_2 :CH3_4 :CH5_6 :CH7_8	
:CH9_10 :CH11_12 :CH13_14 :CH15_16	
[:EVENT]?	
:CONDtion?	
:ENABle	<numeric>
:ENABle?	
:PTRansition	<numeric>
:PTRansition?	
:NTRansition	<numeric>
:NTRansition?	

6.2 Before Reading Detailed Command Descriptions

6.2.1 Tips on Transmitting Commands

The following points should be observed in transmitting commands from this instrument:

- Slot-specific setting

If different units involve the same command, it is necessary to identify the slots in which the units are inserted in transmitting commands. When transmitting a command from this unit to set up a unit, be sure to attach the number of the unit in which the unit is inserted to the subsystem header (except for the DISPlay, SYSTem, and STATus subsystems).

Example:

Slot number	Unit inserted	Command transmitted
Slot1	200M CG	OUTPut1:FREQuency 200000 (Set the frequency of the 200MCG inserted in slot 1)
Slot3	200M 16CH PPG	:SOURce3:PATtern:MODE NORMAl (Set the pattern of the 200M 16CH PPG inserted in slot 3)
Slot4	200M 16CH ED	:SENS44:PATtern:MODE NORMAl (Set the pattern of the 200M 16CH ED inserted in slot 4)

- If a particular parameter setting is shared by a PPG and an ED unit.

If a PPG is inserted in one slot, with an ED unit inserted in another and they share an ON parameter setting, then setting one unit in one way would automatically set the other in a similar way (only if the other has a corresponding setup parameter).

Example:

A 200M 16CH PPG is inserted in slot3, with a 200M 16CH ED inserted in another, and their common pattern setting is ON.

Command	Setting
:SOURce3:PATtern:MODE NORMAl (Pattern mode setting:Parameter shared by the PPG and ED)	Both the PPG and ED units have their pattern mode set to NORMAl.
:SENSe4:PATtern:SYNC:ASYNc:PROGrama1,ON (Auto Sync setting:Parameter limited to the ED)	This parameter is set only for the ED unit without influencing the PPG.

- If a single parameter is set by different commands.

The MP1630B has commands for each unit for starting measurement, etc., even for settings that cannot be performed separately for each unit or slot. Transmitting a command to have effect on one unit will set similar units in the same way.

Example: Where a 200M 16CH PPG is inserted in slot3 and a 200M 16CH ED is inserted in slot4, sets the common setting on of pattern settings for two units.

Command to the PPG inserted in slot 3: :INSTrument3:COUPle:PATtern:SET ON

Command to the ED inserted in slot 4: :INSTrument4:COUPle:PATtern:SET ON

Transmitting either of these commands will set the common setting on.

6.2.2 Buffer Sizes

This instrument provides an output buffer of 16 Kbytes to allow mass data to be setup and queries between the controller and device. The controller should have a buffer large enough to correctly receive responses from the device.

Command-specific buffer space requirements are listed below. For more details on command-specific response size, see the detailed description of the command.

- Reading program data in character strings
Up to 512 bytes.
(Command) : SOURce:PATtern:DATA:CHANnel, etc.
- Reading program data as binary data
Up to 16,384 bytes
(Command) : SOURce:PATtern:BDATA:CHANnel, etc.
- Reading measurement result analysis data
Read the analysis data corresponding to an appointed point of time. The size of data at that point of time can be queried with a command. Allocate a buffer space to meet this size.
(Size read command) :CALCulate:DATA:SIZE?
(Analysis data read command) :CALCulate:DATA:TEXT?
- Other query commands
Up to 256 bytes
The buffer space requirement depends on the command, but a buffer space of 256 bytes would allow successful reading.

6.2.3 How to Read Detailed Command Descriptions

This subsection explains how to read the detailed descriptions of commands that begin with the next section, and points to notes, to aid the reader in understanding them.

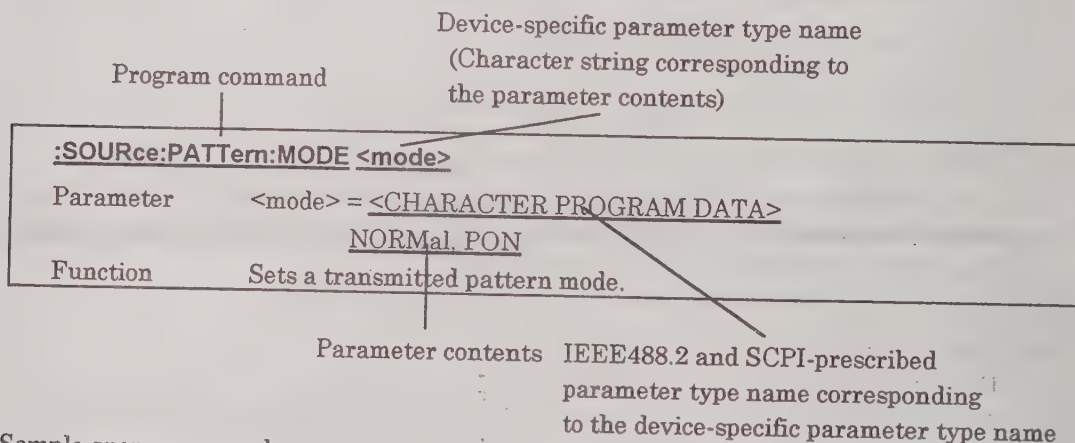
■ Order of presentation of command details

Command details are presented in the order of the commands appearing in the command trees introduced earlier. See Appendix C for the correspondence between the displays or functions and commands.

■ Sample program command

The command shown in the sample below is transmitted in the following form:

:SOURce3:PATtern:MODE NORMal

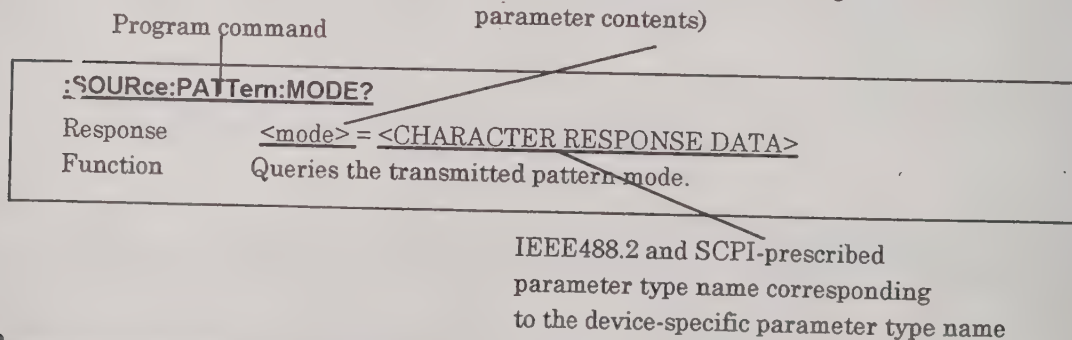


■ Sample query command

A response is omitted when the corresponding program command is available. When the command shown in the sample below is transmitted, a response like that shown below is returned to it:

:SOURce3:PATtern:MODE?

Response : NORM



Note

1. All the commands supported by this instrument are sequential commands.
2. A constraint may be placed on a command when its execution affects any other setting. Refer to the MP1630B Digital Data Analyzer User's Guide and the unit-specific user's guides for the setup parameters that are affected and the conditions that are subject to constraints.

6.3 Setting 200M 16CH PPG (MP163020B)

6.3.1 SOURce Subsystem

The SOURce subsystem mainly sets and queries the following functions:

- Transmitted pattern setting (Normal/PON)
- Error addition setting

Note

Transmit commands by specifying :SOURce3:... if the 200M 16CH PPG is inserted in slot3.

Example: :SOURce3:PATtern:MODE NORMal

:SOURce:PATtern:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
NORMal, PON

Function Sets the transmitted pattern mode.

:SOURce:PATtern:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the transmitted pattern mode.

:SOURce:PATtern[:NORMal]:OMODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
REPeat, BURSt

Function Sets the pattern output mode.

:SOURce:PATtern[:NORMal]:OMODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the pattern output mode.

:SOURce:PATtern[:NORMal]:TYPE <ch>,<type>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
<type> = <CHARACTER PROGRAM DATA>
PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31
PROGram, MIX, ZSUBstitute

Function Sets the pattern type transmitted over a specified CH.

:SOURce:PATtern[:NORMal]:TYPE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the pattern type transmitted over a specified CH.

:SOURce:PATtern[:NORMal]:PRBS:MRATio <ch>,<mratio>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<mratio> = <CHARACTER PROGRAM DATA>

M1_2	1/2
M1_4	1/4
M1_8	1/8
ALL0	ALL0
I1_2	1/2 INVT
M3_4	3/4
M7_8	7/8
ALL1	ALL1

Function Sets the mark ratio when the pattern transmitted over a specified CH is PRBS.

:SOURce:PATtern[:NORMal]:PRBS:MRATio? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <mratio> = <CHARACTER RESPONSE DATA>

Function Queries the mark ratio when the pattern transmitted over a specified CH is PRBS.

:SOURce:PATtern[:NORMal]:PRBS:BSHift <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1	1bit Shift
3	3bit Shift

Function Sets the bit shift when the pattern transmitted over a specified CH is PRBS.

:SOURce:PATtern[:NORMal]:PRBS:BSHift? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit shift when the pattern transmitted over a specified CH is PRBS.

:SOURce:PATtern[:NORMal]:ZSUBstitute:LENGth <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 7 2^7
 9 2^9
 11 2^11
 15 2^15

Function Sets the pattern length when the pattern transmitted over a specified CH is Zero-Subst.

:SOURce:PATtern[:NORMal]:ZSUBstitute:LENGth? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the pattern transmitted over a specified CH is Zero-Subst.

:SOURce:PATtern[:NORMal]:ZSUBstitute:ZLENGth <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 7 to 32767 7 to 32767 bits/Step:1

Sets the length of 0-bit insertion when the pattern transmitted over a specified CH is Zero-Subst.

:SOURce:PATtern[:NORMal]:ZSUBstitute:ZLENGth? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESONSE DATA>

Function Queries the length of 0-bit insertion when the pattern transmitted over a specified CH is Zero-Subst.

:SOURce:PATtern[:NORMal]:ZSUBstitute:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <logic> = <CHARACTER PROGRAM DATA>
 POSitive, NEGative

Function Sets pattern logic when the pattern transmitted over a specified CH is Zero-Subst.

:SOURce:PATtern[:NORMal]:ZSUBstitute:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <logic> = <CHARACTER RESPONSE DATA>

Function Queries the pattern logic when the pattern transmitted over a specified CH is Zero-Subst.

:SOURce:PATtern[:NORMal]:PROGram:LENGth <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 2 to 65536 2~ 8,192bits/Step:1
 8,192~ 65,536bits/Step:8

Function Sets the pattern length when the pattern transmitted over a specified CH is PRGM.

:SOURce:PATtern[:NORMal]:PROGram:LENGth? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the pattern transmitted over a specified CH is PRGM.

:SOURce:PATtern[:NORMal]:PROGram:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <logic> = <CHARACTER PROGRAM DATA>
 POSitive,NEGative

Function Sets pattern logic when the pattern transmitted over a specified CH is PRGM.

:SOURce:PATtern[:NORMal]:PROGram:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <logic> = <CHARACTER RESPONSE DATA>

Function Queries the pattern logic when the pattern transmitted over a specified CH is PRGM.

:SOURce:PATtern[:NORMal]:MIX:PROGram:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <logic> = <CHARACTER PROGRAM DATA>
 POSitive, NEGative

Function Sets the PRGM pattern logic when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:PROGram:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <logic> = <CHARACTER PROGRAM DATA>

Function Queries the PRGM pattern logic when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:PRBS:TYPE <ch>,<type>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<type> = <CHARACTER PROGRAM DATA>

PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31

Function Sets the number of stages of the PRBS block when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:PRBS:TYPE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the number of stages of the PRBS block when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:PRBS:MRATio <ch>,<mratio>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<mratio> = <CHARACTER PROGRAM DATA>

M1_2 1/2

M1_4 1/4

M1_8 1/8

ALL0 ALL0

I1_2 1/2 INVT

M3_4 3/4

M7_8 7/8

ALL1 ALL1

Function Sets the mark ratio when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:PRBS:MRATio? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <mratio> = <CHARACTER RESPONSE DATA>

Function Queries the mark ratio when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:PRBS:BSHift <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 1bit Shift

3 3bit Shift

Function Sets the bit shift of the PRBS block when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:PRBS:BSHift? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit shift of the PRBS block when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:BLOCK:SIZE <ch>,<row>,<column>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<row> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 1 to 32 row/Step:1

<column> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 1 to 8 column/Step:1

Function Sets the number of blocks when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:BLOCK:SIZE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <row> = <NR1 NUMERIC RESPONSE DATA>
 <column> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the number of blocks when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:BLOCK:SET <ch>,<column>,<patt>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <column> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 1 to 8 column/Step:1
 <patt> = <CHARACTER PROGRAM DATA>
 PRBS,PROGram

Function Sets the pattern type of a specified column when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:BLOCK:SET? <ch>,<column>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <column> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 1 to 8 column/Step:1

Response <patt> = <CHARACTER RESPONDS DATA>

Function Queries the pattern type of a specified column when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:BLOCK:LENGTH <ch>,<column>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<column> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 8 1 to 8 column/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

8 to 131072 8 to 131,072 bits/Step:8

Function Sets the block length of a specified column when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:MIX:BLOCK:LENGTH? <ch>,<column>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<column> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 8 1 to 8 column/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the block length of a specified column when the pattern transmitted over a specified CH is MIX.

:SOURce:PATtern[:NORMal]:BURSt:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

INTernal, EXTernal

Function Sets the source of a burst signal.

:SOURce:PATtern[:NORMal]:BURSt:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the source of a burst signal.

:SOURce:PATtern[:NORMal]:BURSt:CYCLe <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0.1 to 10.0 0.1 to 10.0 ms/Step:0.1

Function Sets the burst cycle of a burst signal.

:SOURce:PATtern[:NORMal]:BURSt:CYCLe?

Response <numeric> = <NR2 NUMERIC PROGRAM DATA>

Function Queries the burst cycle of a burst signal.

:SOURce:PATtern[:NORMal]:BURSt:ELENgth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 8 to 2097144 8 to 2,097,144 bits/Step:1

Function Sets the enable length of a burst signal.

:SOURce:PATtern[:NORMal]:BURSt:ELENgth?

Response <numeric> = <NR1 NUMERIC PROGRAM DATA>

Function Queries the enable length of a burst signal.

:SOURce:PATtern[:NORMal]:BURSt:PMODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
 REStart, CONSeCutive

Function Sets the pattern output mode when the transmitted pattern is PRGM or PRGM block of MIX.

:SOURce:PATtern[:NORMal]:BURSt:PMODE?

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the pattern output mode when the transmitted pattern is PRGM or PRGM block of MIX.

:SOURce:PATtern[:NORMal]:PDifference:PHASe <channel>,<phase>

Parameter <channel> = <CHARACTER PROGRAM DATA>
 CH1_16 Specifies CH1...CH16 Phase Difference.
 CH1_8 Specifies CH1...CH8 Phase Difference.
 CH9_16 Specifies CH9...CH16 Phase Difference.

<phase> = <CHARACTER PROGRAM DATA>
 NONE, MUX

Function Sets the phase of each CH in the transmitted pattern when the pattern is PRBS.

:SOURce:PATtern[:NORMal]:PDifference:PHASe? <channel>

Parameter <channel> = <CHARACTER PROGRAM DATA>
 CH1_16 Specifies CH1...CH16 Phase Difference.
 CH1_8 Specifies CH1...CH8 Phase Difference.
 CH9_16 Specifies CH9...CH16 Phase Difference.

Response <phase> = <CHARACTER RESPONSE DATA>

Function Queries the phase of each CH in the transmitted pattern when the pattern is PRBS.

:SOURce:PATtern:PON:ECHannel <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Function Sets the enable CHs when the transmitted pattern is PON.

:SOURce:PATtern:PON:ECHannel?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the enable CHs when the transmitted pattern is PON.

:SOURce:PATtern:PON:PLENgtch <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 0 to 64 0 to 64 bits/Step:1
 Function Sets the preamble length when the transmitted pattern is PON.

:SOURce:PATtern:PON:PLENgtch? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the preamble length when the transmitted pattern is PON.

:SOURce:PATtern:PON:PROGram:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <logic> = <CHARACTER PROGRAM DATA>
 POSitive, NEGative
 Function Sets the logic of the PRGM block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:PROGram:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <logic> = <CHARACTER PROGRAM DATA>
 Function Queries the logic of the PRGM block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:PRBS:TYPE <ch>,<type>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<type> = <CHARACTER PROGRAM DATA>

PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS30

Function Sets the number of stages of the PRBS block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:PRBS:TYPE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the number of stages of the PRBS block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:PRBS:MRATio <ch>,<mratio>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<mratio> = <CHARACTER PROGRAM DATA>

M1_2 1/2

M1_4 1/4

M1_8 1/8

ALL0 ALL0

I1_2 1/2 INVT

M3_4 3/4

M7_8 7/8

ALL1 ALL1

Function Sets the mark ratio of the PRBS block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:PRBS:MRATio? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <mratio> = <CHARACTER RESPONSE DATA>

Function Queries the mark ratio of the PRBS block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:PRBS:BSHift <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 1bit Shift
 3 3bit Shift

Function Sets the bit shift of the PRBS block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:PRBS:BSHift? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit shift of the PRBS block on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:BLOCK:SIZE <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 7 1 to 7 block/Step:1

Function Sets the number of blocks on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:BLOCK:SIZE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the number of blocks on a specified CH when the transmitted pattern is PON.

:SOURce:PATtern:PON:BLOCK:SET <ch>,<blk>,<patt>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<blk> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 7 #1 to #7/Step:1

<patt> = <CHARACTER PROGRAM DATA>

PRBS, PRoGram

Function Sets the pattern type of a specified block when the pattern transmitted over a specified CH is PON.

:SOURce:PATtern:PON:BLOCK:SET? <ch>,<blk>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<blk> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 7 #1 to #7 /Step:1

Response <patt> = <CHARACTER RESPONDS DATA>

Function Queries the pattern type of a specified column when the pattern transmitted over a specified CH is PON.

:SOURce:PATtern:PON:BLOCK:LENGTH <ch>,<blk>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<blk> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 7 #1 to #7 /Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

8 to 131072 8 to 131,072 bits/Step:8

Function Sets the bit length of a specified block when the pattern transmitted over a specified CH is PON.

:SOURce:PATtern:PON:BLOCK:LENGTH? <ch>,<blk>

Parameter <ch> = <DECIMAL NUMERIC PROGRAm DATA>

1 to 16 CH1 to CH16/Step:1

<blk> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 7 #1 to #7/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit length of a specified block when the pattern transmitted over a specified CH is PON.

:SOURce:PATtern:PON:OTIMing:BMODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
INTERNAL, EXTERNAL

Function Sets the source of a burst signal when the transmitted pattern is PON.

:SOURce:PATtern:PON:OTIMing:BMODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the source of a burst signal when the transmitted pattern is PON.

:SOURce:PATtern:PON:OTIMing:BCYCLE <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.1 to 10.0 0.1 to 10.0ms/Step:0.1

Function Sets the burst cycle when the transmitted pattern is PON.

:SOURce:PATtern:PON:OTIMing:BCYCLE?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the burst cycle when the transmitted pattern is PON.

:SOURce:PATtern:PON:OTIMing:GTIME <gt>, <numeric>

Parameter <gt> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 15 GT1 to GT15/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-2097083 to 2097067 -2,097,083 ~ 2,097,067 bits/Step:1

Function Sets the guard time between CHs when the transmitted pattern is PON.

:SOURce:PATtern:PON:OTIMing:GTIME? <gt>

Parameter <gt> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 15 GT1 to GT15/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the guard time between CHs when the transmitted pattern is PON.

:SOURce:PATtern:DATA:CHANnel <ch>,<start>,<end>,<data>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFF 0 to FFFFbit/Step:1 (Starting address: Specified in hex)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFF 0 to FFFFbit/Step:1 (Ending address: Specified in hex)

<data> = <STRING PROGRAM DATA>

"H***..." Specifies pattern data in hex.

***... : 1 to 400 characters (pattern data for 400 x 4 bits);
specifies a character string consisting of any of 0 through 9 and A through F.

"B***..." Specifies pattern data in binary.

***... : 1 to 400 characters (pattern data for 400 bits);
specifies a character string of 0s and 1s.

Function Sets data in the program part of the transmitted pattern between the <start> and <end> addresses when transmitted pattern is PRGM. The specified range of data is overwritten.

If the bit length of the data specified by <data> is less than the range specified by <start> and <end>	Duplicate <data> for the specified range of data. <Examples> • <start>=#H0, <end>=#H20, <data>="HABC" Set data: ABCABCAB • <start>=#H0, <end>=#H8, <data>="B011" Set data: 01101101
If the bit length of the data specified by <data> is greater than the range specified by <start> and <end>	Truncate <data> to the specified range of data. <Examples> • <start>=#H0, <end>=#H10, <data>="HABCDEFGH" Set data: ABCD • <start>=#H0, <end>=#H4, <data>="B01100110" Set data: 0110

Usage example Sets the program pattern in the address range (hex) from 0 to 1F for CH3 to 0.
:SOURce3:PATtern:DATA:CHANnel△3,#H0,#H1F,"H0"

:SOURce:PATtern:DATA:CHANnel? <ch>,<start>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16
	<start> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #HFFFF	0 to FFFFbit/Step:1(Starting address:pecified in hex)
Response	<data> = <STRING RESPONSE DATA>	
	"H***..."	Specifies pattern data in hex. ***... : Up to 400 characters (pattern data for 400×4 bits).
Function	Queries the data in the program part of the transmitted pattern beginning with <start> when tranmitted pattern is PRGM. If pattern data for 400×4 bits is not available, the data from the <start> address to the ending address is returned in response.	

:SOURce:PATtern:DATA:BLOCK <ch>,<blk>,<start>,<end>,<data>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Starting address: pecified in hex)
 (The staring bit of the block is taken as address 0.)
 <end> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Ending address: Specified in hex)
 (The staring bit of the block is taken as address 0.)
 <data> = <STRING PROGRAM DATA>
 "H***..." Specifies pattern data in hex.
 ***... : 1 to 400 characters (pattern data for 400×4 bits);
 specifies a character string consisting of any of 0
 through 9 and A through F.
 "B***..." Specifies pattern data in binary.
 ***... : 1 to 400characters (pattern data for 400 bits);
 specifies a character string of 0s and 1s.

Function Sets program pattern data in a specified block of the transmitted pattern when tranmitted pattern is MIX or PON.

The specified range of data is overwritten.

Usage example Sets the program pattern in the address range (hex) from 0 to 1F for #1 in CH3 to 0.

:SOURce3:PATtern:DATA:BLOCK△3,1,#H0,#HFF,"H0"

:SOURce:PATtern:DATA:BLOCK? <ch>,<blk>,<start>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Starting address:Specified in hex)
 (The staring bit of the block is taken as address 0.)
 Response <data> = <STRING RESPONSE DATA>
 "H***..." Returns pattern data in hex.
 ***... : Up to 400characters (pattern data for 400×4 bits).

Function Queries the program pattern data in the specified block of the transmitted pattern when tranmitted pattern is MIX or PON. Pattern data for 400×4 bits is returned beginning with the <blk> address. If pattern data for 400×4 bits is not available, the data from the <start> address to the ending address in <blk> is returned in response.

:SOURce:PATtern:BDATa:CHANnel <ch>,<start>,<end>,<bdata>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16
	<start> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #HFFFF	0 to FFFFbit/Step:1(Starting address: Specified in hex)
	<end> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #HFFFF	0 to FFFFbit/Step:1(Ending address: Specified in hex)
	<bdata> = <ARBITRARY BLOCK PROGRAM DATA>	
	#YYYY<binary>	X : Number of digits of YYY
		YYY : Byte length of <binary>; 1 to 16,000 (bytes)
		<binary> : Binary data up to 16,000 bytes
Function	Sets data in the program part of the transmitted pattern between the <start> and <end> addresses when transmitted pattern is PRGM. The specified range of data is overwritten.	
Example	Sets the program pattern in the address range (hex) from 0 to 1F for CH3 to 010010010.	
	:SOURce3:PATtern:BDATa:ADDReSS△3,#H0,#HA,#12<binary>	
	<binary> = 2-byte data in binary, consisting of 0100100 and 10XXXXXX (X: 0 or 1).	

:SOURce:PATtern:BDATa:CHANnel? <ch>,<start>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16
	<start> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #HFFFF	0 to FFFFbit/Step:1(Starting address: Specified in hex)
Response	<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>	
	#YYYY<binary>	X : Number of digits of YYY
		YYY : Byte length of <binary>; 1 to 16,000 (bytes)
		<binary> : Binary data up to 16,000 bytes
Function	Queries the data in the program part of the transmitted pattern beginning with <start> when transmitted pattern is PRGM. Pattern data for 16,000 bytes is returned beginning with the <start> address. If pattern data for 16,000 bytes is not available, the data from the <start> address to the ending address is returned in response.	

:SOURce:PATtern:BDATa:BLOCK <ch>,<blk>,<start>,<end>,<bdata>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Starting address: Specified in hex)
 (The starting bit of the block is taken as address 0.)
 <end> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Ending address: Specified in hex)
 (The starting bit of the block is taken as address 0.)
 <bdata> = <ARBITRARY BLOCK PROGRAM DATA>
 #YYYY<binary> X : Number of digits of YYY
 YYY : Byte length of <binary>; 1 to 16,000 (bytes)
 <binary> : Binary data up to 16,000 bytes

Function Sets program pattern data in a specified block of the transmitted pattern when transmitted pattern is MIX or PON. The specified range of data is overwritten.

:SOURce:PATtern:BDATa:BLOCK? <ch>,<blk>,<start>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Starting address: Specified in hex)
 (The starting bit of the block is taken as address 0.)

Response <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
 #YYYY<binary> X : Number of digits of YYY
 YYY : Byte length of <binary>; 1 to 16,000 (bytes)
 <binary> : Binary data up to 16,000 bytes

Function Queries the program pattern data in the specified block of the transmitted pattern when transmitted pattern is MIX or PON. Pattern data for 16,000 bytes is returned beginning with the <start> address. If pattern data for 16,000 bytes is not available, the data from the <start> address to the ending address in <blk> is returned in response.

:SOURce:PATtern:DREVerse:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Reverses the data in the program part of the transmitted pattern when transmitted pattern is PRGM. The range of data that is reversed is specified by CH.

:SOURce:PATtern:DREVerse:BLOCK <ch>,<blk>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1

Function Reverses the data in the program part of the transmitted pattern when transmitted pattern is MIX or PON. The range of data that is reversed is specified by CH and a block number.

:SOURce:PATtern:DREVerse:ADDReSS <ch>,<start>,<end>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1(Starting address: Specified in hex)
 <end> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1(Ending address: Specified in hex)

Function Reverses the data in the program part of the transmitted pattern. The range of data that is reversed is specified by CH and also by a starting address and an ending address.

:SOURce:PATtern:DREVerse:DELTA <ch>,<start>,<delta>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1(Starting address: Specified in hex)
 <delta> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 65536 1 to 65536 bit/Step:1

Function Reverses the data in the program part of the transmitted pattern. The range of data that is reversed is specified by CH and a bit length relative to a starting address.

:SOURce:PATtern:ADDRess:CHANnel? <ch>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1	
Response	<start> = <HEXADECIMAL NUMERIC RESPONSE DATA> #HXXX Returns the starting address in hex.	
	<end> = <HEXADECIMAL NUMERIC RESPONSE DATA> #HYYY Returns the ending address in hex.	
Function	Queries the starting and ending addresses of the program part of the pattern transmitted over a specified CH when transmitted pattern is PRGM.	

:SOURce:PATtern:ADDRess:BLOCK? <ch>,<blk>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1	
	<blk> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 32 #1 to #32/Step:1	
Response	<start> = <HEXADECIMAL NUMERIC RESPONSE DATA> #HXXX Returns the starting address in hex.	
	<end> = <HEXADECIMAL NUMERIC RESPONSE DATA> #HYYY Returns the ending address in hex.	
Function	Queries the starting and ending addresses of a specified PRGM block of the transmitted pattern when transmitted pattern is MIX or PON. This address is absolute address that defined the head bit address of Block1 is equal to 0.	

:SOURce:PATtern:DElay:CYCLe <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 10 1 to 10 sec/Step:1	
Function	Sets the cycle at intervals of which consecutive 32-bit errors are inserted for delay measurement.	

:SOURce:PATtern:DElay:CYCLe?

Response	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
Function	Queries the cycle at intervals of which consecutive 32-bit errors are inserted for delay measurement.	

:SOURce:PATtern:DElay:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>

OFF Do not insert.

DElay1 Insert

Function Sets the insertion of consecutive 32-bit errors for delay measurement ON or OFF. This command is cause to execution error, if a 200M 16CH ED is inserted in Slot4. Where a 200M 16CH ED is inserted in Slot4, sets by following command.
:SENSe:MEASure:TEST<item>

:SOURce:PATtern:DElay:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the ON/OFF status of the insertion of consecutive 32-bit errors for delay measurement.

This command is cause to execution error, if a 200M 16CH ED is inserted in Slot4. Where a 200M 16CH ED is inserted in Slot4, queries by following command.

:SENSe:MEASure:TEST?

:SOURce:PATtern:EADdition:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets the ON/OFF status of error addition.

:SOURce:PATtern:EADdition:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 or 1

Function Queries the ON/OFF status of error addition.

:SOURce:PATtern:EADdition:SINGLE

Function Adds a single error.

:SOURce:PATtern:EADdition:MODE1 <mode1>

Parameter <mode1> = <CHARACTER PROGRAM DATA>
NORMal, BURSt

Function Selects the mode of adding errors; that is, adding errors continuously or adding them in a burst manner.

:SOURce:PATtern:EADdition:MODE1?

Response <mode1> = <CHARACTER PROGRAM DATA>

Function Queries the mode of adding errors; that is, adding errors continuously or adding them in a burst manner.

:SOURce:PATtern:EADdition:MODE2 <mode2>

Parameter <mode2> = <CHARACTER PROGRAM DATA>
INTernal, EXTernal

Function Selects the source of the signal that is used at error addition.

:SOURce:PATtern:EADdition:MODE2?

Response <mode2> = <CHARACTER RESPONSE DATA>

Function Queries the source of the signal that is used at error addition.

:SOURce:PATtern:EADdition:MODE3 <mode3>

Parameter <mode3> = <CHARACTER PROGRAM DATA>
CYCLic, SINGLe, REPeat, LEVel

Function Selects the method of error insertion.
Selectable items depend on setting of MODE1, MODE2.

MODE1	MODE2	MODE3
NORMal	INTernal	CYCLic SINGLe
	EXTnal	None
BURSt	INTernal	REPeat
	EXTnal	LEVel

:SOURce:PATtern:EADdition:MODE3?

Response <mode3> = <CHARACTER RESPONSE DATA>

Function Queries the method of error insertion.

:SOURce:PATtern:EADdition:RATE <ch>, <rate>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1

<rate> = <CHARACTER PROGRAM DATA>

E_2, E_3, E_4, E_5

E_6, E_7, E_8, E_9 ... Correspond to 1E-2, 1E-3, and so on.

Function Selects the rate of errors that are added to a specified CH.

:SOURce:PATtern:EADdition:RATE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1

Response <rate> = <CHARACTER RESPONSE DATA>

Function Queries the rate of errors that are added to a specified CH.

:SOURce:PATtern:EADdition:BCYCLE <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 1 to 10 s/Step:1

Function Sets the burst cycle of burst errors.

:SOURce:PATtern:EADdition:BCYCLE?

Response <numeric> = <NR1 NUMERIC REPOSENSE DATA>

Function Queries the burst cycle of burst errors.

:SOURce:PATtern:EADdition:ELENgth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 20 to 140 20 to 140 ms/Step:20

Function Sets the enable length of burst errors.

:SOURce:PATtern:EADdition:ELENgth?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the enable length of burst errors.

:SOURce:PATtern:EADdition:ADDITION <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Sets the ON/OFF status of error addition to a specified CH.

:SOURce:PATtern:EADdition:ADDITION? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 OFF
 1 ON

Function Queries the ON/OFF status of error addition to a specified CH.

:SOURce:PATtern:EADdition:ABLock <ch>,<mode>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<mode> = <CHARACTER PROGRAM DATA>
 ALL, SELEcted

Function Sets whether the blocks on a specified CH to which errors are added are specified individually or in a group.

:SOURce:PATtern:EADdition:ABLock? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries whether the blocks on a specified CH to which errors are added are specified individually or in a group.

:SOURCE:PATTERN:EADDITION:BLOCK <ch>,<blk>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<blk> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 32 #1 to #32/Step:1

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of error addition to a specified block.

:SOURCE:PATtern:EADDITION:BLOCK? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <blk> = <NR1 NUMERIC RESPONSE DATA>

##...

Returns the blocks to which errors are added, separated by a comma (,).

0

There are no blocks to add errors to.

Function	Queries the ON/OFF status of error addition to a specified block.
----------	---

6.3.2 OUTPut Subsystem

The OUTPut subsystem sets and queries the following functions:

- Clock output interface setting
- Data output interface setting
- AUX output interface setting
- SYNC output setting
- Trigger output setting

Note:

Transmit commands by specifying :OUTPut3:... if the 200M 16CH PPG is inserted in slot3.

Example: :OUTPut3:CLOCK:POLarity 1,POSitive

:OUTPut:SET <boolean>

パラメータ <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

機能 Sets the ON/OFF status of clock output and data output to ALL CHs.

:OUTPut:SET?

レスポンス <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 OFF

1 ON

機能 Queries the ON/OFF status of clock output and data output to ALL CHs.

:OUTPut:CLOCK:POLarity <ch>,<pol>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<pol> = <CHARACTER PROGRAM DATA>

POSitive, NEGative

Function Sets the clock output polarity of a specified CH.

:OUTPut:CLOCK:POLarity? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <pol> = <CHARACTER RESPONSE DATA>

Function Queries the clock output polarity of a specified CH.

:OUTPut:CLOCK:OMODE <ch>,<mode>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <mode> = <CHARACTER PROGRAM DATA>
 REPEAT, BURST

Function Sets the method of clock output to a specified CH.

:OUTPut:CLOCK:OMODE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the method of clock output to a specified CH.

:OUTPut:CLOCK:OUTPut <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of clock output to a specified CH.

:OUTPut:CLOCK:OUTPut? <ch>

Parameter <ch> = <DECIMAL NUMERIC RESPONSE DATA>
 1 to 16 CH1 to CH16/Step:1

Response <NR1 NUMERIC RESPONSE DATA>

0 OFF

1 ON

Function Queries the ON/OFF status of clock output to a specified CH.

:OUTPut:CLOCK:DELaY <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-5.0 to 5.0 -5.0 to 5.0 ns/Step:0.1

Function Sets the clock output delay of a specified CH.

:OUTPut:CLOCK:DELaY? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR2 NUMERIC PROGRAM DATA>

Function Queries the clock output delay of a specified CH.

:OUTPut:CLOCK:LeVel <ch>,<level>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <level> = <CHARACTER PROGRAM DATA>
 VAR, ECL, PECL, TTL, LVTTI
 Function Sets the offset level of clock output for a specified CH.

:OUTPut:CLOCK:LeVel? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <level> = <CHARACTER RESPONSE DATA>
 Function Queries the offset level of clock output for a specified CH.

:OUTPut:CLOCK:TERMination <ch>,<term>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <term> = <CHARACTER PROGRAM DATA>
 GND 50ohm to GND
 V_2 50ohm to -2V
 V3 50ohm to +3V
 OPEN Open
 Function Sets the condition of terminating clock output for a specified CH.

:OUTPut:CLOCK:TERMination? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <term> = <CHARACTER RESPONSE DATA>
 Function Queries the condition of terminating clock output for a specified CH.

:OUTPut:CLOCK:AMPlitude <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 0.50 to 5.00 0.50~5.00Vpp/Step:0.01 (With the DUT termination open)
 0.250 to 2.500 0.250~2.500Vpp/Step:0.005 (otherwise)
 Function Sets the amplitude of clock output for a specified CH.

:OUTPut:CLOCK:AMPLitude? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
 Function Queries the amplitude of clock output for a specified CH.

:OUTPut:CLOCK:OFFSet <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-5.000 to 5.000 -5.000~5.000 V/Step:0.005 (Term....:Open)
-2.5000 to 2.5000 -2.5000~2.5000 V/Step:0.0025(Term....:50ohm to GND)
-3.5000 to 1.5000 -3.5000~5.0000 V/Step:0.0025(Term....:50ohm to -2V)
-1.0000 to 4.0000 0.5000~4.0000 V/Step:0.0025(Term....:50ohm to +3V)
Function Sets the offset value of clock output for a specified CH.

:OUTPut:CLOCK:OFFSet? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries the offset value of clock output for a specified CH.

:OUTPut:CLOCK:OREFERENCE <ch>,<offset>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
<offset> = <CHARACTER PROGRAM DATA>
VOH, VTH, VOL
Function Sets the reference value of clock output for a specified CH.

:OUTPut:CLOCK:OREFERENCE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16
Response <offset> = <CHARACTER RESPONSE DATA>
Function Queries the reference value of clock output for a specified CH.

:OUTPut:DATA:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

DOWNside Downside (To OSU)

UPSide Upside (To ONU)

Function Sets the direction of pattern transmission when the transmitted pattern is PON.

:OUTPut:DATA:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the direction of pattern transmission when the transmitted pattern is PON.

:OUTPut:DATA:OUTPut <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of data output for a specified CH.

:OUTPut:DATA:OUTPut? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <boolean> = <BOOLEAN RESPONSE DATA>

0 OFF

1 ON

Function Queries the ON/OFF status of data output for a specified CH.

:OUTPut:DATA:SKEW <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-5.0 to 5.0 -5.0 to +5.0 ns/Step:0.1

Function Sets the skew of data transmitted over a specified CH.

:OUTPut:DATA:SKEW? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the skew of data transmitted over a specified CH.

:OUTPut:DATA:LEVel <ch>,<level>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<level> = <CHARACTER PROGRAM DATA>
 VAR, ECL, PECL, TTL, LVTTI

Function Sets the offset level of data output for a specified CH.

:OUTPut:DATA:LEVel? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <level> = <CHARACTER RESPONSE DATA>

Function Queries the offset level of data output for a specified CH.

:OUTPut:DATA:TERMination <ch>,<term>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<term> = <CHARACTER PROGRAM DATA>

GND 50ohm to GND

V_2 50ohm to -2V

V3 50ohm to +3V

OPEN Open

Function Sets the condition of terminating data output for a specified CH.

:OUTPut:DATA:TERMination? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <term> = <CHARACTER RESPONSE DATA>

Function Queries the condition of terminating data output for a specified CH.

:OUTPut:DATA:AMPLitude <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0.50 to 5.00 0.50 to 5.00Vpp/Step:0.01 (With the DUT termination open)

0.250 to 2.500 0.250 to 2.500Vpp/Step:0.005 (otherwise)

Function Sets the amplitude of data output for a specified CH.

:OUTPut:DATA:AMPLitude? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the amplitude of data output for a specified CH.

:OUTPut:DATA:OFFSet <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 -5.000 to 5.000 -5.000~5.000 V/Step:0.005 (Term...:Open)
 -2.5000 to 2.5000 -2.5000~2.5000 V/Step:0.0025(Term...:50ohm to GND)
 -3.5000 to 1.5000 -3.5000~5.0000 V/Step:0.0025(Term...:50ohm to -2V)
 -1.0000 to 4.0000 0.5000~4.0000 V/Step:0.0025(Term...:50ohm to +3V)

Function Sets the offset value of data output for a specified CH.

:OUTPut:DATA:OFFSet? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the offset value of data output for a specified CH.

:OUTPut:DATA:OREference <ch>,<offset>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <offset> = <CHARACTER PROGRAM DATA>
 VOH, VTH, VOL

Function Sets the reference value of data output for a specified CH.

:OUTPut:DATA:OREference? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <offset> = <CHARACTER RESPONSE DATA>

Function Queries the reference value of data output for a specified CH.

:OUTPut:AUXiliary:LEVel <level>

Parameter <level> = <CHARACTER PROGRAM DATA>
ECL, TTL

Function Sets the level of all the ports of AUX output.

:OUTPut:AUXiliary:LEVel?

Response <level> = <CHARACTER RESPONSE DATA>

Function Queries the level of all the ports of AUX output.

:OUTPut:AUXiliary:CHANnel <port>,<ch>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 8 Port1 to Port8/Step:1

<ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1

Function Sets the CH to provide output to a specified port.

:OUTPut:AUXiliary:CHANnel? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 8 Port1 to Port8/Step:1

Response <ch> = <NR1 NUMERIC RESPONSE DATA>
Queries the CH to provide output to a specified port.

:OUTPut:AUXiliary:TYPE <port>,<type>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 8 Port1 to Port8/Step:1

<type> = <CHARACTER PROGRAM DATA>
ENvelope, RESet

Function Sets the type of signal that is output to a specified port.

:OUTPut:AUXiliary:TYPE? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 8 Port1 to Port8/Step:1

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the type of signal that is output to a specified port.

:OUTPut:AUXiliary:LOGic <port>,<logic>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 <logic> = <CHARACTER PROGRAM DATA>
 POSitive, NEGative
 Function Sets the logic of a specified port.

:OUTPut:AUXiliary:LOGic? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 Response <logic> = <CHARACTER RESPONSE DATA>
 Function Queries the logic of a specified port.

:OUTPut:AUXiliary:BLOCation <port>,<numeric>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 0 to MAX-1 0 to MAX-1 bits/Step:1
 MAX: Enable length of a burst signal in the transmitted
 pattern
 Function Sets the starting position of output to a specified port.

:OUTPut:AUXiliary:BLOCation? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the starting position of output to a specified port.

:OUTPut:AUXiliary:ELOCation <port>,<numeric>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to MAX 1 to MAX bits/Step:1
 MAX : Enable length of a burst signal in the transmitted
 pattern
 Function Sets the ending position of output to a specified port.

:OUTPut:AUXiliary:ELOCation? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the ending position of output to a specified port.

:OUTPut:AUXiliary:ENvelope:PDElay <port>,<numeric>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 -64 to 9999 -64 to +9999 bits/Step:1

Function Sets the starting position of an envelope signal (delay from the data output starting position) for a specified port.

:OUTPut:AUXiliary:ENvelope:PDElay? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the starting position of an envelope signal (delay from the data output starting position) for a specified port.

:OUTPut:AUXiliary:ENvelope:NDElay <port>,<numeric>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 -64 to 9999 -64 to +9999 bits/Step:1

Function Sets the ending position of an envelope signal (delay from the data output ending position) for a specified port.

:OUTPut:AUXiliary:ENvelope:NDElay? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the ending position of an envelope signal (delay from the data output ending position) for a specified port.

:OUTPut:AUXiliary:RESet:PWIDth <port>,<numeric>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 1 to 16 bits/Step:1

Function Sets the pulse width of the reset signal for a specified port.

:OUTPut:AUXiliary:RESet:PWIDth? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pulse width of the reset signal for a specified port.

:OUTPut:AUXiliary:RESet:LOCation <port>,<loc>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 <loc> = <CHARACTER PROGRAM DATA>
 POSitive Envelope signal rising position
 NEGative Envelope signal falling position
 Function Sets the position of the reset signal for a specified port.

:OUTPut:AUXiliary:RESet:LOCation? <port>

Parameter <port> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 Port1 to Port8/Step:1
 Response <loc> = <CHARACTER PROGRAM DATA>
 Function Queries the position of the reset signal for a specified port.

:OUTPut:SYNC:SOURce <source>

Parameter <source> = <CHARACTER PROGRAM DATA>
 CLOCK1 1/1 Clock Sync
 CLOCK8 1/8 Clock Sync
 PATTERN Pattern Sync
 Function Selects the SYNC Output output signal.

:OUTPut:SYNC:SOURce?

Response <source> = <CHARACTER RESPONSE DATA>
 Function Queries the SYNC Output output signal.

:OUTPut:SYNC:PATTERN:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Function Sets the CH of the pattern with which SYNC output is synchronized when Pattern Sync has been selected as Sync Output.

:OUTPut:SYNC:PATTERN:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the CH of the pattern with which SYNC output is synchronized when Pattern Sync has been selected as Sync Output.

:OUTPut:SYNC:PATTERN:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
 PROGram, PRBS
 Function Sets the type of Pattern with which SYNC output is synchronized.

:OUTPut:SYNC:PATTERN:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>
 Function Queries the type of Pattern with which SYNC output is synchronized.

:OUTPut:TRIGger:SOURce <source>

Parameter <source> = <CHARACTER PROGRAM DATA>
 ERRor Error setting
 BLOCk Block timing setting
 Function Sets the Trigger Output output signal.

:OUTPut:TRIGger:SOURce?

Response <source> = <CHARACTER RESPONSE DATA>
 Function Queries the Trigger Output output signal.

:OUTPut:TRIGger:CHANnel <ch>

Queries the Trigger Output output signal. <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Function Sets the CH to serve as a reference for the Trigger Output output signal.

:OUTPut:TRIGger:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the CH to serve as a reference for the Trigger Output output signal.

:OUTPut:TRIGger:BLOCk <blk>

Parameter <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1
 Function Sets the block to issue an output signal when Block Timing has been specified as Trigger Output.

:OUTPut:TRIGger:BLOCk?

Response <blk> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the block to issue an output signal when Block Timing has been specified as Trigger Output.

:OUTPut:BTRigger:DELa / <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 0 to 9999 0 to 9999 bits/Step:1
 Function Sets the output signal timing of the Burst Trigger Output signal (delay from the pattern enable starting position).

:OUTPut:BTRigger:DELaY? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the output signal timing of the Burst Trigger Output signal (delay from the pattern enable starting position).

6.3.3 CALCulate Subsystem

The CALCulate subsystem queries the status of alarm occurrence (PPG Clock Loss).

Note

Transmit commands by specifying :CALCulate3:... if the 200M 16CH PPG is inserted in slot3.

Example: :CALCulate3:DATA:MONitor?

:CALCulate:DATA:MONitor?

Response	<STRING RESPONSE DATA>	
	"Occur"	An alarm is present.
	(NL^END)	No alarm is present.
	"-----"	No data is available.
Function	Queries the status of PPG Clock Loss.	

6.3.4 INSTRUMENT Subsystem

The INSTRUMENT subsystem sets and queries the common slot settings and groupings.

Note

Transmit commands by specifying :INSTRUMENT3:... if the 200M 16CH PPG is inserted in slot3.

Example: :INSTRUMENT3:COUPLE:PATTERN:SET ON

:INSTRUMENT:COUPLE:PATTERN:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0 Common slot 3 and 4 settings OFF

ON or 1 Common slot 3 and 4 settings ON

Function Sets the common setting ON/OFF status of transmitted pattern settings for two units.

:INSTRUMENT:COUPLE:PATTERN:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 OFF

1 ON

Function Queries the common setting ON/OFF status of transmitted pattern settings for two units.

6.4 Setting 200M 16CH ED (MP163040B)

6.4.1 SENSE Subsystem

The SENSE subsystem sets and queries the following functions:

- Received pattern setting
- Error/alarm measurement setting
- Delay measurement setting
- Eye margin setting
- Measurement result analysis setting

Note

Transmit commands by specifying :SENSe4:... if the 200M 16CH ED is inserted in slot4.

Example: :SENSe4:PATtern:MODE NORMal

:SENSe:PATtern:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
NORMal, PON

Function Sets the received pattern mode.

:SENSe:PATtern:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the received pattern mode.

:SENSe:PATtern[:NORMal]:IMODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
REPeat, BURSt

Function Sets the pattern input mode.

:SENSe:PATtern[:NORMal]:IMODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the pattern input mode.

:SENSe:PATtern[:NORMal]:TYPE <ch>,<type>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <type> = <CHARACTER PROGRAM DATA>
 PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31
 PROGram, MIX, ZSUBstitute
 Function Sets the pattern type received over a specified CH.

:SENSe:PATtern[:NORMal]:TYPE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <type> = <CHARACTER RESPONSE DATA>
 Function Queries the pattern type received over a specified CH.

:SENSe:PATtern[:NORMal]:PRBS:MRATio <ch>,<mratio>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <mratio> = <CHARACTER PROGRAM DATA>
 M1_2 1/2
 M1_4 1/4
 M1_8 1/8
 ALL0 ALL0
 I1_2 1/2 INVT
 M3_4 3/4
 M7_8 7/8
 ALL1 ALL1
 Function Sets the mark ratio when the pattern received over a specified CH is PRBS.

:SENSe:PATtern[:NORMal]:PRBS:MRATio? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <mratio> = <CHARACTER RESPONSE DATA>
 Function Queries the mark ratio when the pattern received over a specified CH is PRBS.

:SENSe:PATtern[:NORMal]:PRBS:BSHift <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 1bit Shift
 3 3bit Shift

Function Sets the bit shift when the pattern received over a specified CH is PRBS.

:SENSe:PATtern[:NORMal]:PRBS:BSHift? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit shift when the pattern received over a specified CH is PRBS.

:SENSe:PATtern[:NORMal]:ZSUBstitute:LENGth <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 7 2⁷
 9 2⁹
 11 2¹¹
 15 2¹⁵

Function Sets the pattern length when the pattern received over a specified CH is Zero-Subst.

:SENSe:PATtern[:NORMal]:ZSUBstitute:LENGth? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the pattern received over a specified CH is Zero-Subst.

:SENSe:PATtern[:NORMal]:ZSUBstitute:ZLENgth <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

7 to 32767 3 to 32767 bits/Step:1

Function Sets the length of 0-bit insertion when the pattern received over a specified CH is Zero-Subst.

:SENSe:PATtern[:NORMal]:ZSUBstitute:ZLENgth? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESONSE DATA>

Function Queries the length of 0-bit insertion when the pattern received over a specified CH is Zero-Subst.

:SENSe:PATtern[:NORMal]:ZSUBstitute:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<logic> = <CHARACTER PROGRAM DATA>

POSitive, NEGative

Function Sets pattern logic when the pattern received over a specified CH is Zero-Subst.

:SENSe:PATtern[:NORMal]:ZSUBstitute:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <logic> = <CHARACTER PROGRAM DATA>

Function Queries the pattern logic when the pattern received over a specified CH is Zero-Subst.

:SENSe:PATtern[:NORMal]:PROGram:LENgth <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

2 to 65536 2~ 8,192bits/Step:1

8,192~ 65,536bits/Step:8

Function Sets the pattern length when the pattern received over a specified CH is PRGM.

:SENSe:PATtern[:NORMal]:PROGram:LENgth? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the pattern received over a specified CH is PRGM.

:SENSe:PATtern[:NORMal]:PROGram:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <logic> = <CHARACTER PROGRAM DATA>
 POSitive,NEGative

Function Sets pattern logic when the pattern received over a specified CH is PRGM.

:SENSe:PATtern[:NORMal]:PROGram:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <logic> = <CHARACTER RESPONSE DATA>

Function Queries the pattern logic when the pattern received over a specified CH is PRGM.

:SENSe:PATtern[:NORMal]:MIX:PROGram:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <logic> = <CHARACTER PROGRAM DATA>
 POSitive, NEGative

Function Sets the PRGM pattern logic when the pattern received over a specified CH is PRGM.

:SENSe:PATtern[:NORMal]:MIX:PROGram:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <logic> = <CHARACTER PROGRAM DATA>

Function Queries the PRGM pattern logic when the pattern received over a specified CH is PRGM.

:SENSe:PATtern[:NORMal]:MIX:PRBS:TYPE <ch>,<type>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <type> = <CHARACTER PROGRAM DATA>
 PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31

Function Sets the number of stages of the PRBS block when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:PRBS:TYPE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the number of stages of the PRBS block when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:PRBS:MFATio <ch>,<mratio>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<mratio> = <CHARACTER PROGRAM DATA>

M1_2 1/2

M1_4 1/4

M1_8 1/8

ALL0 ALL0

I1_2 1/2 INVT

M3_4 3/4

M7_8 7/8

ALL1 ALL1

Function Sets the mark ratio when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:PRBS:MRATio? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <mratio> = <CHARACTER RESPONSE DATA>

Function Queries the mark ratio when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:PRBS:BSHift <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 1bit Shift

3 3bit Shift

Function Sets the bit shift of the PRBS block when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:PRBS:BSHift? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit shift of the PRBS block when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:BLOCK:SIZE <ch>,<row>,<column>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<row> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 32 1 to 32 row/Step:1

<column> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 8 1 to 8 column/Step:1

Function Sets the number of blocks when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:BLOCK:SIZE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <row> = <NR1 NUMERIC RESPONSE DATA>

<column> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the number of blocks when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:BLOCK:SET <ch>,<column>,<patt>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<column> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 8 1 to 8 column/Step:1

<patt> = <CHARACTER PROGRAM DATA>

PRBS,PROGram

Function Sets the pattern type of a specified column when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:BLOCK:SET? <ch>,<column>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<column> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 8 1 to 8 column/Step:1

Response <patt> = <CHARACTER RESPONSE DATA>

Function Queries the pattern type of a specified column when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:BLOCK:LENGth <ch>,<column>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <column> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 1 to 8 column/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 8 to 131072 8 to 131,072 bits/Step:8

Function Sets the block length of a specified column when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:MIX:BLOCK:LENGth? <ch>,<column>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <column> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 8 1 to 8 column/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the block length of a specified column when the pattern received over a specified CH is MIX.

:SENSe:PATtern[:NORMal]:SYNC:ASYNc:PROGram <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Sets the automatic synchronization ON/OFF status of the PRGM pattern.

:SENSe:PATtern[:NORMal]:SYNC:ASYNc:PROGram? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 OFF
 1 ON

Function Queries the automatic synchronization ON/OFF status of the PRGM pattern.

:SENSe:PATtern[:NORMal]:SYNC:ASYNc:PRBS <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Set the automatic synchronization ON/OFF status of the PRBS pattern.

:SENSe:PATtern[:NORMal]:SYNC:ASYNc:PRBS? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 OFF
 1 ON

Function Queries the automatic synchronization ON/OFF status of the PRBS pattern.

:SENSe:PATtern[:NORMal]:SYNC:THReshold:PROGram <ch>,<thre>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <thre> = <CHARACTER PROGRAM DATA>
 E_2, E_3, E_4, E_5, E_6, E_7, E_8, AUTO
 Correspond to 1E-2, 1E-3, and so on.

Function Sets a pattern sync loss threshold level for the PRGM pattern.

:SENSe:PATtern[:NORMal]:SYNC:THReshold:PROGram? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <thre> = <CHARACTER RESPONSE DATA>

Function Queries the pattern sync loss threshold level for the PRGM pattern.

:SENSe:PATtern[:NORMal]:SYNC:THReshold:PRBS <ch>,<thre>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <thre> = <CHARACTER PROGRAM DATA>
 E_2, E_3, E_4, E_5, E_6, E_7
 Correspond to 1E-2, 1E-3, and so on.

Function Sets a pattern sync loss threshold level for the PRBS pattern.

:SENSe:PATtern[:NORMal]:SYNC:THReshold:PRBS? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <thre> = <CHARACTER RESPONSE DATA>

Function Queries the pattern sync loss threshold level for the PRBS pattern.

:SENSe:PATtern[:NORMal]:SYNC:PSMode <ch>,<mode>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <mode> = <CHARACTER PROGRAM DATA>
 NORMal, FRAMe

Function Sets the method of PRGM pattern synchronization.

:SENSe:PATtern[:NORMal]:SYNC:PSMode? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the method of PRGM pattern synchronization.

:SENSe:PATtern[:NORMal]:SYNC:BSMode <ch>,<mode>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <mode> = <CHARACTER PROGRAM DATA>
 EXTeRnal, INTeRnal

Function Sets the method of burst synchronization.

:SENSe:PATtern[:NORMal]:SYNC:BSMode? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <mode> = <CHARACTER REPOSENSE DATA>

Function Queries the method of burst synchronization.

:SENSe:PATtern[:NORMal]:SYNC:FLENgth <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 8, 16, 24, 32 Correspond to 8, 16, 23, and 32 bits, respectively.

Function Sets the frame length when the Frame method of synchronization has been selected.

:SENSe:PATtern[:NORMal]:SYNC:FLENgth? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the frame length when the Frame method of synchronization has been selected.

:SENSe:PATtern[:NORMal]:BURSt:ELENgth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

8 to 2097144 8 to 2,097,144 bytes/Step:1

Function Sets the enable length of a burst received pattern.

:SENSe:PATtern[:NORMal]:BURSt:ELENgth?

Response <numeric> = <NR1 NUMERIC PROGRAM DATA>

Function Queries the enable length of a burst received pattern.

:SENSe:PATtern[:NORMal]:BURSt:PMODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

REStart, CONSeCutive

Function Sets the pattern input mode when the received pattern is PRGM or PRGM block of MIX.

:SENSe:PATtern[:NORMal]:BURSt:PMODE?

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the pattern input mode when the received pattern is PRGM or PRGM block of MIX.

:SENSe:PATtern:PON:ECHannel <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
Function Sets the enable CHs when the received pattern is PON.

:SENSe:PATtern:PON:ECHannel?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 16 CH1 to CH16
Function Queries the enable CHs when the received pattern is PON.

:SENSe:PATtern:PON:PROGram:LOGic <ch>,<logic>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
<logic> = <CHARACTER PROGRAM DATA>
POSitive, NEGative
Function Sets the logic of the PRGM block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:PROGram:LOGic? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
Response <logic> = <CHARACTER PROGRAM DATA>
Function Queries the logic of the PRGM block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:PRBS:TYPE <ch>,<type>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
<type> = <CHARACTER PROGRAM DATA>
PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS30
Function Sets the number of stages of the PRBS block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:PRBS:TYPE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
Response <type> = <CHARACTER RESPONSE DATA>
Function Queries the number of stages of the PRBS block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:PRBS:MRATio <ch>,<mratio>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<mratio> = <CHARACTER PROGRAM DATA>

M1_2 1/2

M1_4 1/4

M1_8 1/8

ALL0 ALL0

I1_2 1/2 INVT

M3_4 3/4

M7_8 7/8

ALL1 ALL1

Function Sets the mark ratio of the PRBS block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:PRBS:MRATio? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <mratio> = <CHARACTER RESPONSE DATA>

Function Queries the mark ratio of the PRBS block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:PRBS:BSHift <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 1bit Shift

3 3bit Shift

Function Sets the bit shift of the PRBS block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:PRBS:BSHift? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

:Similar to <numeric>

in :SENSe:PATtern:PON:PRBS:BSHift.

Function Queries the bit shift of the PRBS block on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:BLOCK:SIZE <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 7 1 to 7 block/Step:1

Function Sets the number of blocks on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:BLOCK:SIZE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 1 to 7 1 to 7

Function Queries the number of blocks on a specified CH when the received pattern is PON.

:SENSe:PATtern:PON:BLOCK:SET <ch>,<blk>,<patt>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 7 #1 to #7/Step:1

<patt> = <CHARACTER PROGRAM DATA>
 PRBS PRBS pattern
 PROGram PRGM pattern

Function Sets the pattern type of a specified block when the received pattern is PON.

:SENSe:PATtern:PON:BLOCK:SET? <ch>,<blk>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

<blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 7 #1 to #7 /Step:1

Response <patt> = <CHARACTER RESPONSe DATA>

Function Queries the pattern type of a specified column when the received pattern is PON.

:SENSe:PATtern:PON:BLOCK:LENGth <ch>,<blk>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16 / Step:1
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 7 #1 to #7 / Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 8 to 131072 8 to 131,072 bits / Step:8

Function Sets the bit length of a specified block when the received pattern is PON.

:SENSe:PATtern:PON:BLOCK:LENGth? <ch>,<blk>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16 / Step:1
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 7 #1 to #7 / Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit length of a specified block when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:ASYNc:PROGram <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16 / Step:1
 <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, On or 1

Function Sets the automatic synchronization ON/OFF status of the PRGM pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:ASYNc:PROGram? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16 / Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 OFF
 1 ON

Function Queries the automatic synchronization ON/OFF status of the PRGM pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:ASYNc:PRBS <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Set the automatic synchronization ON/OFF status of the PRBS pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:ASYNc:PRBS? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 or 1

Function Queries the automatic synchronization ON/OFF status of the PRBS pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:THReshold:PROGram <ch>,<thre>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <thre> = <CHARACTER PROGRAM DATA>
 E_2, E_3, E_4, E_5, E_6, E_7, E_8, AUTO
 . Correspond to 1E-2, 1E-3, and so on.

Function Sets an pattern sync loss threshold level for the PRGM pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:THReshold:PROGram? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <thre> = <CHARACTER RESPONSE DATA>

Function Queries the pattern sync loss threshold level for the PRGM pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:THReshold:PRBS <ch>,<thre>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<thre> = <CHARACTER PROGRAM DATA>

E_2, E_3, E_4, E_5, E_6, E_7

Correspond to 1E-2, 1E-3, and so on.

Function Sets an pattern sync loss threshold level for the PRBS pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:THReshold:PRBS? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <thre> = <CHARACTER RESPONSE DATA>

Function Queries the pattern sync loss threshold level for the PRBS pattern when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:BSMode <ch>,<mode>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<mode> = <CHARACTER PROGRAM DATA>

EXTernal, INTernal

Function Sets the method of burst synchronization when the p received pattern is PON.

:SENSe:PATtern:PON:SYNC:BSMode? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <mode> = <CHARACTER REPONSE DATA>

Function Queries the method of burst synchronization when the received pattern is PON.

:SENSe:PATtern:PON:SYNC:FLENgth <ch>,<numeric>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1 <numeric> = <DECIMAL NUMERIC PROGRAM DATA> 8, 16, 24, 32 Correspond to 8, 16, 23, and 32 bits, respectively.
Function	Sets the frame length when the received pattern is PON and the Frame method of synchronization has been selected.

:SENSe:PATtern:PON:SYNC:FLENgth? <ch>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Queries the frame length when the received pattern is PON and the Frame method of synchronization has been selected.

:SENSe:PATtern:DATA:CHANnel <ch>,<start>,<end>,<data>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFF 0 to FFFFbit/Step:1(Starting address: Specified in hex)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFF 0 to FFFFbit/Step:1(Ending address: Specified in hex)

<data> = <STRING PROGRAM DATA>

"H*** ..." Specifies pattern data in hex.

***... : 1 to 400 characters (pattern data for 400 × 4 bits);

specifies a character string consisting of any of 0 through 9 and A through F.

"B*** ..." Specifies pattern data in binary.

***... : 1 to 400 characters (pattern data for 400 bits);

specifies a character string of 0s and 1s.

Function Sets data in the program part of the received pattern between the <start> and <end> addresses when received pattern is PRGM. The specified range of data is overwritten.

If the bit length of the data specified by <data> is less than the range specified by <start> and <end>	Duplicate <data> for the specified range of data. <Examples> • <start>=#H0, <end>=#H20, <data>="HABC" Set data : ABCABCAB • <start>=#H0, <end>=#H8, <data>="B011" Set data : 01101101
If the bit length of the data specified by <data> is greater than the range specified by <start> and <end>	Truncate <data> to the specified range of data. <Examples> • <start>=#H0, <end>=#H10, <data>="HABCDEFGH" Set data : ABCD • <start>=#H0, <end>=#H4, <data>="B01100110" Set data : 0110

Example Sets the program pattern in the address range (hex) from 0 to 1F for CH3 to 0.

:SENSe4:PATtern:DATA:CHANnel△3,#H0,#H1F,"H0"

:SENSe:PATtern:DATA:CHANnel? <ch>,<start>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16
	<start> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #HFFFF	0 to FFFFbit/Step:1(Starting address: Specified in hex)
Response	<data> = <STRING RESPONSE DATA>	
	"H***..."	Specifies pattern data in hex. ***... : Up to 400 characters (pattern data for 400×4 bits).
Function	Queries the data in the program part of the received pattern beginning with <start> when received pattern is PRGM. If pattern data for 400×4 bits is not available, the data from the <start> address to the ending address is returned in response.	

:SENSe:PATtern:DATA:BLOCK <ch>,<blk>,<start>,<end>,<data>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16
	<blk> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 32	#1 to #32/Step:1
	<start> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #H1FFF	0 to 1FFFbit/Step:1(Starting address: Specified in hex) (The starting bit of the block is taken as address 0.)
	<end> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #H1FFF	0 to 1FFFbit/Step:1(Ending address: Specified in hex) (The starting bit of the block is taken as address 0.)
	<data> = <STRING PROGRAM DATA>	
	"H***..."	Specifies pattern data in hex.
	***...	: 1 to 400 characters (pattern data for 400×4 bits); specifies a character string consisting of any of 0 through 9 and A through F.
	"B***..."	Specifies pattern data in binary.
	***...	: 1 to 400 characters (pattern data for 400 bits); specifies a character string of 0s and 1s.
Function	Sets program pattern data in a specified block of the received pattern when received pattern is MIX or PON. The specified range of data is overwritten.	
Example	Sets the program pattern in the address range (hex) from 0 to 1F for #1 in CH3 to 0. :SENSe4:PATtern:DATA:BLOCK△3,1,#H0,#HFF,"H0"	

:SENSe:PATtern:DATA:BLOCK? <ch>,<blk>,<start>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16
	<blk> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 32	#1 to #32/Step:1
	<start> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #H1FFF	0 to 1FFFbit/Step:1(Starting address: Specified in hex) (The starting bit of the block is taken as address 0.)
Response	<data> = <STRING RESPONSE DATA>	
	"H***..."	Returns pattern data in hex.
	***...	: Up to 400 characters (pattern data for 400×4 bits).
Function	Queries the program pattern data in the specified block of the received pattern when received pattern is MIX or PON. Pattern data for 400×4 bits is returned beginning with the <blk> address. If pattern data for 400×4 bits is not available, the data from the <start> address to the ending address in <blk> is returned in response.	

:SENSe:PATtern:BDATa:CHANnel <ch>,<start>,<end>,<data>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1(Starting address: Specified in hex)
 <end> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1(Ending address: Specified in hex)
 <bdata> = <ARBITRARY BLOCK PROGRAM DATA>
 #YYYY<binary> X : Number of digits of YYY
 YYY : Byte length of <binary>; 1 to 16,000 (bytes)
 <binary> : Binary data up to 16,000 bytes

Function Sets data in the program part of the received pattern between the <start> and <end> addresses when received pattern is PRGM. The specified range of data is overwritten.

:SENSe:PATtern:BDATa:CHANnel? <ch>,<start>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1

Response <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
 #YYYY<binary> X : Number of digits of YYY
 YYY : Byte length of <binary>; 1 to 16,000 (bytes)
 <binary> : Binary data up to 16,000 bytes

Function Queries the data in the program part of the received pattern beginning with <start> when received pattern is PRGM. Pattern data for 16,000 bytes is returned beginning with the <start> address. If pattern data for 16,000 bytes is not available, the data from the <start> address to the ending address is returned in response.

:SENSe:PATtern:BDATa:BLOCK <ch>,<blk>,<start>,<end>,<data>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Starting address: Specified in hex)
 (The starting bit of the block is taken as address 0.)
 <end> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Ending address: Specified in hex)
 (The starting bit of the block is taken as address 0.)
 <bdata> = <ARBITRARY BLOCK PROGRAM DATA>
 #YYYY<binary> X : Number of digits of YYY
 YYY : Byte length of <binary>; 1 to 16,000 (bytes)
 <binary> : Binary data up to 16,000 bytes

Function Sets program pattern data in a specified block of the received pattern when
 received pattern is MIX or PON. The specified range of data is overwritten.

:SENSe:PATtern:BDATa:BLOCK? <ch>,<blk>,<start>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #H1FFF 0 to 1FFFbit/Step:1(Starting address: Specified in hex)
 (The starting bit of the block is taken as address 0.)

Response <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
 #YYYY<binary> X : Number of digits of YYY
 YYY : Byte length of <binary>; 1 to 16,000 (bytes)
 <binary> : Binary data up to 16,000 bytes

機能 Queries the program pattern data in the specified block of the received pattern
 when received pattern is MIX or PON. Pattern data for 16,000 bytes is returned
 beginning with the <start> address. If pattern data for 16,000 bytes is not
 available, the data from the <start> address to the ending address in <blk> is
 returned in response.

:SENSe:PATtern:DREVerse:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Reverses the data in the program part of the received pattern when received pattern is PRGM. The range of data that is reversed is specified by CH.

:SENSe:PATtern:DREVerse:BLOCK <ch>,<blk>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1

Function Reverses the data in the program part of the received pattern when received pattern is MIX or PON. The range of data that is reversed is specified by CH and a block number.

:SENSe:PATtern:DREVerse:ADDReSS <ch>,<start>,<end>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1(Starting address: Specified in hex)
 <end> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFFbit/Step:1(Ending address: Specified in hex)

Function Reverses the data in the program part of the received pattern. The range of data that is reversed is specified by CH and also by a starting address and an ending address.

:SENSe:PATtern:DREVerse:DELTA <ch>,<start>,<delta>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <start> = <NON-DECIMAL PROGRAM DATA>
 #H0 to #HFFFF 0 to FFFF bit/Step:1(Starting address: Specified in hex)
 <delta> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 65536 0 to 65536 bit/Step:1

Function Reverses the data in the program part of the received pattern. The range of data that is reversed is specified by CH and a bit length relative to a starting address.

:SENSe:PATtern:ADDReSS:CHANnel? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <start> = <HEXADECIMAL NUMERIC RESPONSE DATA>
 #HXXX Returns the starting address in hex.
 <end> = <HEXADECIMAL NUMERIC RESPONSE DATA>
 #HYYY Returns the ending address in hex.

Function Queries the starting and ending addresses of the program part of the PRGM pattern received over a specified CH.

:SENSe:PATtern:ADDReSS:BLOCK? <ch>,<blk>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1

Response <start> = <HEXADECIMAL NUMERIC RESPONSE DATA>
 #HXXX Returns the starting address in hex.
 <end> = <HEXADECIMAL NUMERIC RESPONSE DATA>
 #HYYY Returns the ending address in hex.

Function Queries the starting and ending addresses of a specified PRGM block of the received pattern when received pattern is MIX or PON.
 This address is absolute address that defined the head bit address of Block1 is equal to 0.

:SENSe:MEASure:TEST <item>

Parameter	<item> = <CHARACTER PROGRAM DATA>	
	EALarm	Error/Alarm
	EMARgin	Eye Margin
	DELay1	Delay(Mode1)
	DELay2	Delay(Mode2)

Function Sets a measurement item.

Where a 200M 16CH PPG is inserted in Slot3, consecutive 32-bit errors for Delay(Mode1) measurement is inserted automatically in PPG by sets DELay1.

This command is cause to execution error, if a 200M 16CH ED is not inserted in Slot4. In this case, sets insertion of consecutive 32-bit errors for delay measurement ON or OFF by following command.

:SOURce:PATtern:DELay:TYPE<type>

:SENSe:MEASure:TEST?

Response	<item> = <CHARACTER RESPONSE DATA>
Function	Queries the measurement item.

:SENSe:MEASure:START

Function Starts measurement.

:SENSe:MEASure:STOP

Function Stops measurement.

:SENSe:MEASure:EALarm:MODE <mode>

Parameter	<mode> = <CHARACTER PROGRAM DATA>	
	REPeat, SINGLE, UNTimed	

Function Sets the mode of error/alarm measurement.

:SENSe:MEASure:EALarm:MODE?

Response	<mode> = <CHARACTER RESPONSE DATA>
Function	Queries the mode of error/alarm measurement.

:SENSe:MEASure:EALarm:PERiod <day>,<hour>,<min>,<sec>

Parameter <day>,<hour>,<min>,<sec> = <DECIMAL NUMERIC PROGRAM DATA>

 <day> = 0 to 99 0~99 日/Step:1

 <hour> = 0 to 23 0~23 時/Step:1

 <min> = 0 to 59 0~59 分/Step:1

 <sec> = 0 to 59 0~59 秒/Step:1

Function Sets the measurement period.

:SENSe:MEASure:EALarm:PERiod?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement period.

:SENSe:MEASure:EALarm:BTIME:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

 OFF or 0, ON or 1

Function Sets the ON/OFF status of the function that sets measurement start times.

:SENSe:MEASure:EALarm:BTIME:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

 0 OFF

 1 ON

Function Queries the ON/OFF status of the function that sets measurement start times.

:SENSe:MEASure:EALarm:BTIME:START <year>,<month>,<day>,<hour>,<min>,<sec>

Parameter <year>,<month>,<day>,<hour>,<min>,<sec> =

 <DECIMAL NUMERIC PROGRAM DATA>

Function Sets the measurement start time.

:SENSe:MEASure:EALarm:BTIME:START?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =

 <NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement start time.

:SENSe:MEASure:EALarm:ERRor:MODE <ch>,<mode>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1	
	<mode> = <CHARACTER PROGRAM DATA>	
	OFF	OFF
	ALL	ALL
	INSection	INS
	OMISsion	OMI
	IOmmission	INS & OMI

Function Sets the mode of error detection on a specified CH.

:SENSe:MEASure:EALarm:ERRor:MODE? <ch>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1
-----------	---

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the mode of error detection on a specified CH.

:SENSe:MEASure:EALarm:ERRor:DBLock <ch>,<mode>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1	
	<mode> = <CHARACTER PROGRAM DATA> ALL SElected	

Function Sets whether blocks are individually specified for the purpose of detecting errors on a specified CH or all blocks are specified.

:SENSe:MEASure:EALarm:ERRor:DBLock? <ch>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1
-----------	---

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries whether blocks are individually specified for the purpose of detecting errors on a specified CH or all blocks are specified.

:SENSe:MEASure:EALarm:ERRor:BLOCK <ch>,<blk>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <blk> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1~#32/Step:1
 <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Sets whether error detection in specified blocks is enabled.

:SENSe:MEASure:EALarm:ERRor:BLOCK? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
Response <blk> = <NR1 NUMERIC RESPONSE DATA>
 #,#,... Returns the numbers of the blocks in which error detection is
 enabled, separated by a comma (,).

Function Queries the numbers of the blocks in which error detection is enabled.

:SENSe:MEASure:EALarm:STARt?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
 <NR1 NUMERIC RESPONSE DATA>

Function Queries the error/alarm measurement start time. 0,0,0,0,0 is returned when
 there is no error/alarm measurement start time.

:SENSe:MEASure:EALarm:STOP?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
 <NR1 NUMERIC RESPONSE DATA>

Function Queries the error/alarm measurement stop time. 0,0,0,0,0 is returned when
 there is no error/alarm measurement stop time.
 Measurement stop time can be buffered with the :CALCulate:DATA:STORe
 command. Buffered stop time provides a response to queries.

:SENSe:MEASure:EALarm:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 Measurement is halted.
 1 Measurement is in progress.

Function Queries the status of error/alarm measurement in progress.

:SENSe:MEASure:EALarm:ELAPsed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the elapsed time of error/alarm measurement.

:SENSe:MEASure:EALarm:TIMed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the remaining time of error/alarm measurement.

:SENSe:MEASure:EALarm:ITime?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function Queries the time at which the intermediate data during error/alarm measurement was created.

The time at which the intermediate data can be buffered with the :CALCulate:DATA:STORe command. Buffered time provides a response to queries.

:SENSe:MEASure:DELAy1:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
 REPeat, SINGle

Function Sets the delay (mode 1) measurement mode.

:SENSe:MEASure:DELAy1:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the mode of delay (mode 1) measurement.

:SENSe:MEASure:DELAy1:TOJT <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0.5 0.5s

1 1s

2 2s

5 5s

10 10s

Function Sets the delay (mode 1) measurement timeout period.

:SENSe:MEASure:DELAy1:TOUT?

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the delay (mode 1) measurement timeout period.

:SENSe:MEASure:DELAy1:CHANnel <ch>,<boolean>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

 1 to 16 CH1 to CH16/Step:1

 <boolean> = <BOOLEAN PROGRAM DATA>

 OFF or 0, ON or 1

Function Sets a CH that is subject to delay (mode 1) measurement.

:SENSe:MEASure:DELAy1:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

 #,#,... Returns the CHs that have been selected, separated by a
 comma (,).

 0 All CHs are OFF.

Function Queries the CHs that are subject to delay (mode 1) measurement.

:SENSe:MEASure:DELAy1:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 Measurement is halted.

1 to 16 Measurement is in progress (the executing CH is returned).

Function Queries the status of delay (mode 1) measurement in progress.

:SENSe:MEASure:DELAy1:MStAtE? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 Halted

1 Awaiting

2 Measuring

Function Queries the status of delay (mode 1) measurement on a specified CH in progress.

:SENSe:MEASure:DELAy1:STARt?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =

<NR1 NUMERIC RESPONSE DATA>

Function Queries the delay (mode 1) measurement start time. 0,0,0,0,0.0 is returned when there is no measurement start time.

:SENSe:MEASure:DELAy1:STOP?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =

<NR1 NUMERIC RESPONSE DATA>

When there is no delay (mode 1) measurement stop time

Function Queries the delay (mode 1) measurement stop time. 0,0,0,0,0.0 is returned when there is no measurement stop time.

Measurement stop time can be buffered with the :CALCulate:DATA:STORe command. Buffered stop time provides a response to queries.

:SENSe:MEASure:DELAy1:ELAPsed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the elapsed time of delay (mode 1) measurement.

:SENSe:MEASure:DELaY1:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>

 TIME Time unit measurement

 BIT Bit unit measurement

Function Sets the Delay(Mode1) measurement method.

:SENSe:MEASure:DELaY1:TYPE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the Delay(Mode1) measurement method.

:SENSe:MEASure:DELAy2:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
 REPeat, SINGLE

Function Sets the delay (mode 2) measurement mode.

:SENSe:MEASure:DELAy2:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the mode of delay (mode 2) measurement.

:SENSe:MEASure:DELAy2:TOUT <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0.5	0.5s
1	1s
2	2s
5	5s
10	10s

Function Sets the delay (mode 2) measurement timeout period.

:SENSe:MEASure:DELAy2:TOUT?

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the delay (mode 2) measurement timeout period.

:SENSe:MEASure:DELAy2:PATtern <patt>

Parameter <patt> = <CHARACTER PROGRAM DATA>
 PRBS7, PRBS15, PRBS23, PRBS31

Function Sets the number of stages of the PRBS pattern used for delay (mode 2) measurement.

:SENSe:MEASure:DELAy2:PATtern?

Response <patt> = <CHARACTER RESPONSE DATA>

Function Queries the number of stages of the PRBS pattern used for delay (mode 2) measurement.

:SENSe:MEASure:DELay2:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Function Sets a CH that is subject to delay (mode 2) measurement.

:SENSe:MEASure:DELay2:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the CHs that are subject to delay (mode 2) measurement.

:SENSe:MEASure:DELay2:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 Measurement is halted.

1 Measurement is in progress

Function Queries the status of delay (mode 2) measurement in progress. "In progress" refers to the period of time until the Stop button is pressed or measurement is otherwise stopped after the Start button was pressed.

:SENSe:MEASure:DELay2:MState?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 Halted

1 Awaiting

2 Measuring

Function Queries the status of delay (mode 2) measurement in progress.

:SENSe:MEASure:DELay2:START?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =

<NR1 NUMERIC RESPONSE DATA>

Function Queries the delay (mode 2) measurement start time. 0,0,0,0,0,0 is returned when there is no measurement start time.

:SENSe:MEASure:DELay2:STOP?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =

<NR1 NUMERIC RESPONSE DATA>

Function Queries the delay (mode 2) measurement stop time. 0,0,0,0,0,0 is returned when there is no measurement stop time.

Measurement stop time can be buffered with the :CALCulate:DATA:STORe command. Buffered stop time provides a response to queries.

:SENSe:MEASure:DELaY2:ELAPsed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>
 Function Queries the elapsed time of delay (mode 2) measurement.

:SENSe:MEASure:DELaY2:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
 TIME Time unit measurement
 BIT Bit unit measurement
 Function Sets the Delay(Mode2) measurement method.

:SENSe:MEASure:DELaY2:TYPE?

Response <mode> = <CHARACTER RESPONSE DATA>
 Function Queries the Delay(Mode2) measurement method.

:SENSe:MEASure:EMARgin:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
 MARGIn, DIAGram

Function Sets the eye margin measurement mode.

:SENSe:MEASure:EMARgin:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the eye margin measurement mode.

:SENSe:MEASure:EMARgin:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>

 PTHReshold Phase & Threshold

 PHASe hase

 THReshold Threshold

 POINT8 points

 POINT16 16points

 POINT32 32points

 POINT64 64points

Function Sets an eye margin measurement item or a measurement point at the same error rate.

:SENSe:MEASure:EMARgin:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the eye margin measurement item or measurement point at the same error rate.

:SENSe:MEASure:EMARgin:MARGIn:THReshold <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>

 E_3, E_4, E_5, E_6, E_7, E_8, E_9

 Correspond to 1E-2, 1E-3,..., and 1E-9, respectively.

Function Sets an error rate in the Margin measurement mode.

:SENSe:MEASure:EMARgin:MARGIn:THReshold?

Response <thre> = <CHARACTER REPONSE DATA>

Function Queries the error rate in the Margin measurement mode.

:SENSe:MEASure:EMARgin:MARGin:RESolution <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
COARse, FINE

Function Sets an error rate resolution in the Margin measurement mode.

:SENSe:MEASure:EMARgin:MARGin:RESolution?

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the error rate resolution in the Margin measurement mode.

:SENSe:MEASure:EMARgin:DIAGram:THReshold <thre>,<boolean>

Parameter <thre> = <CHARACTER PROGRAM DATA>
E_3, E_4, E_5, E_6, E_7, E_8, E_9
Correspond to 1E-3, 1E-4,..., and 1E-9, respectively.
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets an error rate in the Diagram measurement mode.

:SENSe:MEASure:EMARgin:DIAGram:THReshold?

Response <thre> = <CHARACTER RESPONSE DATA>
E_#,E_#,... Returns the error rates that have been selected,
separated by a comma (.).

Function Queries the error rate in the Diagram measurement mode.

:SENSe:MEASure:EMARgin:DIAGram:POINT? <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>
E_3, E_4, E_5, E_6, E_7, E_8, E_9
Correspond to 1E-3, 1E-4,..., and 1E-9, respectively.

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
X,Y,Z,... Valid points are returned.
X,Y,Z,...: 1 to 64

0 <thre> is invalid or there are no valid points.

Function Queries the valid measurement points at each error rate in the Diagram measurement mode.

:SENSe:MEASure:EMARgin:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot3
 4 Unit inserted in slot4

Function Sets an ED unit that is subject to eye margin measurement.

:SENSe:MEASure:EMARgin:UNIT?

Response <unit> = <NR1 NUMERIC REPONSE DATA>

Function Queries the ED unit that is subject to eye margin measurement.

:SENSe:MEASure:EMARgin:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Sets a CH that is subject to eye margin measurement.

:SENSe:MEASure:EMARgin:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>
 1 to 16 CH1 to CH16

Function Queries the CHs that are subject to eye margin measurement.

:SENSe:MEASure:EMARgin:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 Measurement is halted.
 1 Measurement is in progress.
 -1 Measurement has failed.

Function Queries the status of progress of eye margin measurement.

:SENSe:MEASure:EMARgin:START?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
 <NR1 NUMERIC RESPONSE DATA>

Function Queries the eye margin measurement start time. 0,0,0,0,0.0 is returned when there is no measurement start time.

:SENSe:MEASure:EMARgin:STOP?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
 <NR1 NUMERIC RESPONSE DATA>

Function Queries the eye margin measurement stop time. 0,0,0,0,0.0 is returned when there is no measurement stop time.

:SENSe:MEASure:EMARgin:ELAPsed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the elapsed time of eye margin measurement.

:SENSe:MEASure:ASEarch:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

TPHase	Threshold & Phase
THReshold	Threshold
PHASe	Phase
OFF	OFF

Function Sets the auto search operation mode.

:SENSe:MEASure:ASEarch:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the auto search operation mode.

:SENSe:MEASure:ASEarch:CHANnel <channel>,<boolean>

Parameter <channel> = <CHARACTER PROGRAM DATA>

CH1,	CH1
...	
CH16	CH16
ALL	ALL CHs

 <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0,	ON or 1
-----------	---------

Function Sets the auto search CH.

:SENSe:MEASure:ASEarch:CHANnel?

Response <channel> = <CHARACTER RESPONSE DATA>

CH#,CH#,..	Returns the CHs that have been selected, separated by a comma (,).
------------	--

NONE	All CHs are OFF.
------	------------------

Function Queries the auto search CH.

:SENSe:MEASure:ASEarch:STARt

Function Starts an auto search.

:SENSe:MEASure:ASEarch:STOP

Function Forces an auto search to a stop.

:SENSe:MEASure:ASEarch:PATtern <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of a PRBS pattern search.

:SENSe:MEASure:ASEarch:PATtern?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 or 1

Function Queries the ON/OFF status of a PRBS pattern search.

:SENSe:MEASure:ASEarch:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 Measurement is halted.

1 to 16 Measurement is in progress (the executing CH is returned).

-1 The auto search has failed.

Function Queries the status of the auto search (executing CH) in progress.

:SENSe:MEASure:ASEarch:FCHannel?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

X,Y,Z,.. The failing CH is returned.

X,Y,Z,...:1 to 16

0 The auto search has not failed.

Function Queries the CH failing in the auto search.

:SENSe:MEASure:RDlagram:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>
 1 to 16 CH1 to CH16
 0 Data is Invalid

Function Queries the CHs that are subject to Recall Eye Diagram data.

:SENSe:MEASure:RDlagram:THReshold?

Response <thre> = <CHARACTER RESPONSE DATA>
 E_#,E_#,... Returns the error rates that have been selected,
 separated by a comma (,).
 NONE Data is Invalid

Function Queries the error rate that are subject to Recall Eye Diagram data.

:SENSe:MEASure:RDlagram:POINt? <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>
 E_3, E_4, E_5, E_6, E_7, E_8, E_9
 Correspond to 1E-3, 1E-4,..., and 1E-9, respectively.

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 X,Y,Z,... Valid points are returned.
 X,Y,Z,...: 1 to 64
 0 <thre> is invalid or there are no valid points.

Function Queries the valid measurement points at each error rate in the Recall Eye Diagram data.

:SENSe:MEASure:RDlagram:START?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
 <NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement start time of Recall Eye Diagram data. 0,0,0,0,0.0 is returned when there is no measurement start time.

:SENSe:MEASure:RDlagram:STOP?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
 <NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement stop time of Recall Eye Diagram data. 0,0,0,0,0.0 is returned when there is no measurement stop time.

:SENSe:ANALyze:CHANnel <channel>,<boolean>

Parameter <channel> = <CHARACTER PROGRAM DATA>

CH1	CH1
:	
CH16	CH16
ALL	Total ALL Error
INSertion	Total INS Error(INS/OMI)
OMISsion	Total OMI Error(INS/OMI)
IOMission	Total INS&OMI Error

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets CHs that are subject to analysis.

:SENSe:ANALyze:CHANnel?

Response <channel> = <CHARACTER RESPONSE DATA>

 #,#,... Returns the CHs that have been selected, separated by a comma (,).

Function Queries the CHs that are subject to analysis.

:SENSe:ANALyze:TTEXT:DATA <data>,<boolean>

Parameter <data> = <CHARACTER PROGRAM DATA>

START	Start Data
INTERmediate	Intermediate Data
END	End Data
SECond	1s Data
ALARm	Alarm Occur Recover Time

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the data items to be collected as trace text data.

:SENSe:ANALyze:TTEXT:DATA?

Response <data> = <CHARACTER RESPONSE DATA>

 XXX,XXX,... Returns the data items that have been selected, separated by a comma (,).

 NONE None have been selected.

Function Queries the data items that are collected as trace text data.

:SENSe:ANALyze:TTEXT:IDATa:ITEM <item>,<boolean>

Parameter <item> = <CHARACTER PROGRAM DATA>

ERRor	Error
THReshold	Threhsold EI/%EFI
PCOunt	Performance Count
PPERcent	Performance %
ALARm	Alarm Interval
FREQuency	Clock Frequency Data

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the data items to be collected as intermediate data.

:SENSe:ANALyze:TTEXT:IDATa:ITEM?

Response <item> = <CHARACTER RESPONSE DATA>

 XXX,XXX,.. Returns the data items that have been selected,
 separated by a comma (.).

 NONE None have been selected.

Function Queries the data items that are collected as intermediate data.

:SENSe:ANALyze:TTEXT:IDATa:PERiod <numeric>,<suffix>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

 1, 10, 15, 60

 <suffix> = <CHARACTER PROGRAM DATA>

 s Second (s)

 m Minute (min)

Function Sets the cycle at intervals of which intermediate data is collected.

:SENSe:ANALyze:TTEXT:IDATa:PERiod?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

 <suffix> = <CHARACTER PROGRAM DATA>

Function Queries the cycle at intervals of which intermediate data is collected.

:SENSe:ANALyze:TTEXT:EDATa:ITEM <item>,<boolean>

Parameter <item> = <CHARACTER PROGRAM DATA>

ERRor	Error
THReshold	Threhsold EI/%EFI
PCOunt	Performance Count
PPERcent	Performance %
ALARm	Alarm Interval
FREQuency	Clock Frequency Data

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the data items to be collected as ending data.

:SENSe:ANALyze:TTEXT:EDATa:ITEM?

Response <item> = <CHARACTER RESPONSE DATA>

XXX,XXX,... Returns the data items that have been selected,
separated by a comma (,).

NONE None have been selected.

Function Queries the data items that are collected as final data.

:SENSe:ANALyze:TTEXT:SDATa:THReshold <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>

NONE None

E_9, E_8, E_7, E_6, E_5, E_4, E_3, E_2

Correspond to 1E-9, 1E-8,..., and 1E-2, respectively.

Function Sets an error data threshold level for collecting 1 second data.

:SENSe:ANALyze:TTEXT:SDATa:THReshold?

Parameter <thre> = <CHARACTER PROGRAM DATA>

Function Queries the error data threshold level for collecting 1 second data.

6.4.2 INPut Subsystem

The INPut subsystem sets and queries the following functions:

- Clock input interface setting
- Data input interface setting
- Reference clock setting

Note

Transmit commands by specifying :INPut4:... if the 200M 16CH ED is inserted in slot4

Example: :INPut4:CLOCK:POLarity 1,POSitive

:INPut:CLOCK:MODE <mode>

Parameter	<mode> = <CHARACTER PROGRAM DATA>	
	NORMal	Normal
	COMMon	Common(Clock1)
	REFerence	Common(Reference)
Function	Sets the clock input mode.	

:INPut:CLOCK:MODE?

Response	<mode> = <CHARACTER RESPONSE DATA>
Function	Queries the clock input mode.

:INPut:CLOCK:IMODE <ch>,<mode>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16/Step:1
	<mode> = <CHARACTER PROGRAM DATA>	
	REPeat, BURSt	
Function	Sets the method of clock input to a specified CH.	

:INPut:CLOCK:IMODE? <ch>

Parameter	<ch> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 16	CH1 to CH16/Step:1
Response	<mode> = <CHARACTER REPOSENSE DATA>	
Function	Queries the method of clock input to a specified	

:INPut:CLOCK:POLarity <ch>,<pol>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <pol> = <CHARACTER PROGRAM DATA>
 POSive,NEGative
 Function Sets the clock input polarity of a specified CH.

:INPut:CLOCK:POLarity? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <pol> = <CHARACTER RESPONSE DATA>
 Function Queries the clock input polarity of a specified CH.

:INPut:CLOCK:DElay <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 -5.0 to 5.0 -5.0 to 5.0 ns/Step:0.1
 Function Sets a clock input delay for a specified CH.

:INPut:CLOCK:DElay? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <numeric> = <NR2 NUMERIC PROGRAM DATA>
 -5.0 to 5.0 -5.0 to 5.0 ns/Step:0.1
 Function Queries the clock input delay for a specified CH.

:INPut:CLOCK:TERMination <ch>,<term>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <term> = <CHARACTER PROGRAM DATA>
 GND GND
 V_2 -2V
 V3 +3V
 Function Sets the condition of terminating clock input for a specified CH.

:INPut:CLOCK:TERMination? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 Response <term> = <CHARACTER RESPONSE DATA>
 Function Queries the condition of terminating clock input for a specified CH.

:INPut:DATA:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
 DOWNside Downside(From ONU)
 UPSide Upside(From OSU)

Function Sets the direction of receiving the received pattern when it is PON.

:INPut:DATA:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the direction of receiving the received pattern when it is PON.

:INPut:DATA:LEVel <ch>,<level>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <level> = <CHARACTER PROGRAM DATA>
 VAR, ECL, PECL, TTL, CMOS, LVTTI

Function Sets the offset level of data input for a specified CH.

:INPut:DATA:LEVel? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <level> = <CHARACTER RESPONSE DATA>

Function Queries the offset level of data input for a specified CH.

:INPut:DATA:THReshold <ch>,<numeric>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 -5.000 to 5.000 -5.000 to +5.000 V/Step:0.005

Function Sets the threshold level of data input for a specified CH.

:INPut:DATA:THReshold? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 -5.000 to 5.000

Function Queries the threshold level of data input for a specified CH.

:INPut:DATA:TERMination <ch>,<term>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1
 <term> = <CHARACTER PROGRAM DATA>
 GND GND
 V_2 -2V
 V3 +3V

Function Sets the condition of terminating data input for a specified CH.

:INPut:DATA:TERMination? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <term> = <CHARACTER RESPONSE DATA>

Function Queries the condition of terminating data input for a specified CH.

:INPut:RClock:SOURce <source>

Parameter <source> = <CHARACTER PROGRAM DATA>
 RCLock Reference Clock Input Port
 CLOCk1 Clock1 Input Port

Function Sets the reference clock when the received pattern is PON.

:INPut:RClock:SOURce?

Response <source> = <CHARACTER RESPONSE DATA>

Function Queries the reference clock when the received pattern is PON.

6.4.3 CALCulate Subsystem

The CALCulate subsystem sets and queries the following functions:

- Error/alarm measurement condition setting
- Measurement result reading
- Measurement result analysis data reading

Note

Transmit commands by specifying :CALCulate4:... if the 200M 16CH ED is inserted in slot4.

Example: :CALCulate4:EALarm:CLEValuation ON

:CALCulate:EALarm:CLEValuation <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets whether the Clock Loss period is subject to measurement evaluation (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:CLEValuation?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 OFF
1 ON

Function Queries whether the Clock Loss period is subject to measurement evaluation (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:SLEValuation <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets whether the Sync Loss period is subject to measurement evaluation (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:SLEValuation?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 OFF
1 ON

Function Queries whether the Sync Loss period is subject to measurement evaluation (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:ERRor:INterval <numeric>,<suffix>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 500
	<suffix> = <CHARACTER PROGRAM DATA> ms, s

Function	Sets an interval of EI and %EFI measurement. Only <10,ms>, <500,ms>, and <1,s> can be set.
----------	--

:CALCulate:EALarm:ERRor:INTerval?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 <suffix> = <CHARACTER RESPONSE DATA>

Function	Queries the interval of EI and %EFI measurement.
----------	--

:CALCulate:EALarm:PERFomance:MEASurement <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function	Sets the ON/OFF status of error performance measurement.
----------	--

:CALCulate:EALarm:PERFomance:MEASurement?

Parameter	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 : OFF
	1 : ON

Function	Queries the ON/OFF status of error performance measurement.
----------	---

:CALCulate:EALarm:PERformance:THReshold <thre>

```
Parameter      <thre> = <STRING PROGRAM DATA>
                "SES_3:DM_6"  SES:1E-3/DM:1E-6
                "SES_4:DM_8"  SES:1E-4/DM:1E-8
```

Function	Sets the threshold level for SES or DM calculation at performance measurement.
----------	--

:CALCulate:EALarm:PERFomance:THReshold?

Response <thre> = <STRING RESPONSE DATA>

Function	Queries the threshold level for SES or DM calculation at performance measurement.
----------	---

:CALCulate:EALarm:TEInterval:MEASurement <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of threshold EI/%EFI measurement.

:CALCulate:EALarm:TEInterval:MEASurement?

Response <numeric> = <NR1 NUMERIC RESPNSE DATA>

0 OFF

1 ON

Function Queries the ON/OFF status of threshold EI/%EFI measurement.

:CALCulate:EALarm:PFail <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of Power Fail measurement.

:CALCulate:EALarm:PFail?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 OFF

1 ON

Function Queries the ON/OFF status of Power Fail measurement.

:CALCulate:DATA:EALarm? <channel>,<string>

Parameter <channel> = <CHARACTER PROGRAM DATA>
 CH1 to CH16 CH1 to CH16/Step:1
 TOTal Total
 Specify <TOTal> to read total errors.
 Specify <CH1> to read power failures and frequencies.
 <string> = <STRING PROGRAM DATA>
 "[CURRent:]<result>" Current data
 "LAST:<result>" Measurement end data on the last
 measurement performed
 "INTermediate:<result>" Intermediate measurement data
 For the contents of <result> see Table 6-2.

Response <string> = <STRING RESPONSE DATA>

Type	Format	Explanation
Form1	" XXXXXXXX"	0 to 9999999
Integer type	"X.XXXXXEEX"	1.0000E07 to 9.9999E16
	"-----"	There is no data for the query.
Form2	"X.XXXXXE-XX"	0.0000E-16 to 1.0000E00
Decimal type	"-----"	There is no data for the query.
	"-----"	There is no data for the query.
Form3	"XXX.XXXXX"	0.0000 to 100.0000
% type	"-----"	There is no data for the query.
	"-----"	There is no data for the query.
Form4	"XXXXXXXXXX"	00 to MAX (Hz)
Frequency type	"-----"	There is no data for the query.

Function Queries the measurement data specified by the parameter. Measurement end data and intermediate measurement data can be buffered with the :CALCulate:DATA:STORe command. Buffered measurement end data or intermediate measurement data provides a response to queries for measurement end data or intermediate measurement data.

Example

- 1) Query the current data on Error Rate (INS) for CH1.
:CALCulate4:DATA:EALarm?△1,"CURRent:ER:INS"
or :CALCulate4:DATA:EALarm?△1,"ER:INSertion" (CURRent: omissible)
Response:"1.0000E-02"
- 2) Query the measurement end data on Cock Loss for CH2.
:CALCulate4:DATA:EALarm?△2,"LAST:AINTErval:CLOSS"
Response:" 123"
- 3) Query the intermediate measurement data on power failures.
:CALCulate4:DATA:EALarm?△1,"INTermediate:AINTErval:POWer"
Response:" 22"

Query the current data on Error Rate (INS) for Total ALL Error.
:CALCulate4:DATA:EALarm?△TOTal,"ER:ALL"
Response:"1.0000E-03"

Table 6-2 Error/Alarm Measurement Items <result> (1/3)

Item			<resul >	Response
Error Rate	INS		"ER:INSeRtion"	Form2
	OMI		"ER:OMISsion"	
	ALL		"ER:ALL"	
	INS&OMI(INS)		"ER:IOMission:INSeRtion"	
	INS&OMI(OMI)		"ER:IOMission:OMISsion"	
Error Count	INS		"EC:INSeRtion"	Form1
	OMI		"EC:OMISsion"	
	ALL		"EC:ALL"	
	INS&OMI(INS)		"EC:IOMission:INSeRtion"	
	INS&OMI(OMI)		"EC:IOMission:OMISsion"	
EI	INS		"EI:INSeRtion"	Form1
	OMI		"EI:OMISsion"	
	ALL		"EI:ALL"	
%EFI	INS		"EFI:INSeRtion"	Form3
	OMI		"EFI:OMISsion"	
	ALL		"EFI:ALL"	
Threshold EI	INS	>1E-3	"THRehosld:EI:INSeRtion:E_3"	Form1
		:	:	
		>1E-8 =<1E-8	"THRehosld:EI:INSeRtion:E_8" "THRehosld:EI:INSeRtion:UE_8"	
	OMI	>1E-3	"THRehosld:EI:OMISsion:E_3"	
		:	:	
		>1E-8 =<1E-8	"THRehosld:EI:OMISsion:E_8" "THRehosld:EI:OMISsion:UE_8"	
	ALL	>1E-3	"THRehosld:EI:ALL:E_3"	
		:	:	
		>1E-8 =<1E-8	"THRehosld:EI:ALL:E_8" "THRehosld:EI:ALL:UE_8"	
Threshold %EFI	INS	>1E-3	"THRehosld:EFI:INSeRtion:E_3"	Form3
		:	:	
		>1E-8 =<1E-8	"THRehosld:EFI:INSeRtion:E_8" "THRehosld:EFI:INSeRtion:UE_8"	
	OMI	>1E-3	"THRehosld:EFI:OMISsion:E_3"	
		:	:	
		>1E-8 =<1E-8	"THRehosld:EFI:OMISsion:E_8" "THRehosld:EFI:OMISsion:UE_8"	
	ALL	>1E-3	"THRehosld:EFI:ALL:E_3"	
		:	:	
		>1E-8 =<1E-8	"THRehosld:EFI:ALL:E_8" "THRehosld:EFI:ALL:UE_8"	

Table 6-2 Error/Alarm Measurement Items <result> (2/3)

Item			<result>	Response
G.821	%ES	INS	"G821:ES:INSertion"	Form3
		OMI	"G821:ES:OMISsion"	
		ALL	"G821:ES:ALL"	
	%EFS	INS	"G821:EFS:INSertion"	
		OMI	"G821:EFS:OMISsion"	
		ALL	"G821:EFS:ALL"	
	%SES	INS	"G821:SES:INSertion"	
		OMI	"G821:SES:OMISsion"	
		ALL	"G821:SES:ALL"	
	%US	INS	"G821:US:INSertion"	
		OMI	"G821:US:OMISsion"	
		ALL	"G821:US:ALL"	
	%DM	INS	"G821:DM:INSertion"	
		OMI	"G821:DM:OMISsion"	
		ALL	"G821:DM:ALL"	
	ES	INS	"G821:ES2:INSertion"	Form1
		OMI	"G821:ES2:OMISsion"	
		ALL	"G821:ES2:ALL"	
	EFS	INS	"G821:EFS2:INSertion"	
		OMI	"G821:EFS2:OMISsion"	
		ALL	"G821:EFS2:ALL"	
	SES	INS	"G821:SES2:INSertion"	
		OMI	"G821:SES2:OMISsion"	
		ALL	"G821:SES2:ALL"	
	US	INS	"G821:US2:INSertion"	
		OMI	"G821:US2:OMISsion"	
		ALL	"G821:US2:ALL"	
	DM	INS	"G821:DM2:INSertion"	
		OMI	"G821:DM2:OMISsion"	
		ALL	"G821:DM2:ALL"	
	EC	INS	"G821:EC2:INSertion"	
		OMI	"G821:EC2:OMISsion"	
		ALL	"G821:EC2:ALL"	

Table 6-2 Error/Alarm Measurement Items <result> (3/3)

Item		<result>	Response
Average Error Rate per second	INS	"AVERAge:ER:INSertion"	Form2
	OMI	"AVERAge:ER:OMISSion"	
	ALL	"AVERAge:ER:ALL"	
Average Error Count per second	INS	"AVERAge:EC:INSertion"	Form1
	OMI	"AVERAge:EC:OMISSion"	
	ALL	"AVERAge:EC:ALL"	
Frequency		"FREQuency"	Form4
Power Fail		"AINTerval:POWer"	Form1
Clock Loss		"AINTerval:CLOSS"	Form1
Pattern Sync Loss	PRGM	"AINTerval:PATtern:PROGram"	Form1
	PRBS	"AINTerval:PATtern:PRBS"	

:CALCulate:DATA:EMARgin? <string>

Parameter <string> = <STRING PROGRAM DATA>

<result> Measurement data

For the contents of <result> see Table 6-3.

Response

<string> = <STRING RESPONSE DATA>

Type	Format	Explanation
Form1	" XX.X"	0 to MAX (ns)
Phase type	"Failed"	Measurement failure
	"-----"	There is no data for the query.
Form2	" XXX"	0 to MAX (mVpp)
Threshold type	"Failed"	Measurement failure
	"-----"	There is no data for the query.
Form3	"\$X.X:\$Y.YYY"	\$: Sign
Eye map type		X.X : Phase(ns)
		Y.YYY : Threshold Voltage(V)

Function

Queries the eye margin measurement data specified by the parameter.

Example

- 1) Query phase margin measurement data.

:CALCulate4:DATA:EMARgin?△"PHASe"

Response:" 4.5"

- 2) Query the value of point 1 in the 1E-4 eye map.

:CALCulate4:DATA:EMARgin?△"MAP:E_4:POINT1"

Response:"-3.7:-0.200"

Table 6-3 Eye Margin Measurement Items <result>

Item		<result>	Response format
Phase Margin		"PHASe"	Form1
Threshold Margin		"THReshold"	Form2
Eye Map	1E-3	"MAP:E_3:POINt1",...,"MAP:E_3:POINt64"	Form3
	1E-4	"MAP:E_4:POINt1",...,"MAP:E_4:POINt64"	
	1E-5	"MAP:E_5:POINt1",...,"MAP:E_5:POINt64"	
	1E-6	"MAP:E_6:POINt1",...,"MAP:E_6:POINt64"	
	1E-7	"MAP:E_7:POINt1",...,"MAP:E_7:POINt64"	
	1E-8	"MAP:E_8:POINt1",...,"MAP:E_8:POINt64"	
	1E-9	"MAP:E_9:POINt1",...,"MAP:E_9:POINt64"	
		Returns the value of measurement data at each point at each error rate.	
	1E-3	"RDIagram:MAP:E_3:POINt1",...,"RDIagram:MAP:E_3:POINt64"	
	1E-4	"RDIagram:MAP:E_4:POINt1",...,"RDIagram:MAP:E_4:POINt64"	
	1E-5	"RDIagram:MAP:E_5:POINt1",...,"RDIagram:MAP:E_5:POINt64"	
	1E-6	"RDIagram:MAP:E_6:POINt1",...,"RDIagram:MAP:E_6:POINt64"	
	1E-7	"RDIagram:MAP:E_7:POINt1",...,"RDIagram:MAP:E_7:POINt64"	
	1E-8	"RDIagram:MAP:E_8:POINt1",...,"RDIagram:MAP:E_8:POINt64"	
	1E-9	"RDIagram:MAP:E_9:POINt1",...,"RDIagram:MAP:E_9:POINt64"	
		Returns the value of recall eye diagram data at each point at each error rate.	

:CALCulate:DATA:DElay1? <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Response <string> = <STRING RESPONSE DATA>

"XXXXXXXXXXXX" 0 to MAX (ms)

"XXXXXXXXXXXX" 0 to MAX (bit)

"Time out " Timeout

"Failed " Measurement failure

"-----" There is no data for the query.

Function Queries the Delay(Mode1) measurement data (delay time) on a specified CH.
 When setting of measurement method is TIME, measurement data is us unit.
 When setting of measurement method is BIT, measurement data is bit unit.
 The Delay(Mode1) measurement data can be buffered with the
 :CALCulate:DATA:STORe command. Buffered measurement data provides a
 response to queries.

:CALCulate:DATA:DElay2?

Response <string> = <STRING RESPONSE DATA>

"XXXXXXXXXXXX" 0 to MAX (ms)

"XXXXXXXXXXXX" 0 to MAX (bit)

"Time out " Timeout

"Failed " Measurement failure

"-----" There is no data for the query.

Function Queries the Delay(Mode2) measurement data (delay time) on a specified CH.
 When setting of measurement method is TIME, measurement data is us unit.
 When setting of measurement method is BIT, measurement data is bit unit.
 The Delay(Mode2) measurement data can be buffered with the
 :CALCulate:DATA:STORe command. Buffered measurement data provides a
 response to queries.

:CALCulate:DATA:MONitor? <ch>,<item>

Parameter	<channel> = <CHARACTER PROGRAM DATA>	
	CH1 to CH16 CH1 to CH16/Step:1	
	<item> = <STRING PROGRAM DATA>	
	"BIT:INSertion"	Bit Error(INS Error)
	"BIT:OMISsion"	Bit Error(OMI Error)
	"BIT:ALL"	Bit Error(ALL Error)
	"BIT:IOMission:INSertion"	Bit Error(INS&OMI Error(INS))
	"BIT:IOMission:OMISsion"	Bit Error(INS&OMI Error(OMI))
	"PATtern:PROGram"	Pattern Sync Loss PRGM
	"PATtern:PRBS"	Pattern Sync Loss PRBS
	"POWer"	Power Fail
	"ED:CLOSs"	ED Clock Loss
	"FEMpty"	FIFO Empty
	"FFULl"	FIFO FULL
	"PERRor"	Phase Error
Response	Specify <1> as <ch> when <item> is <"POWER"> or <"ED:CLOSs"> or <"FEMpty"> or <"FFULl"> or <"PERRor">.	
	<string> = <STRING RESPONSE DATA>	
	"Occur"	An alarm is present.
	(NL^END)	No alarm is present.
	"-----"	There is no data for the query.
Function	Queries the status of occurrence of the monitor items specified by the parameter.	

:CALCulate:DATA:STARt? <type>

Parameter	<type> = <CHARACTER PROGRAM DATA>	
	TTEXT	Trace text data
Response	RTEXT	Recall text data
	<year>,<month>,<day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>	
Function	Queries the start time of the Trace Text data or Recall Text data stored in analysis data 0,0,0,0,0 is returned when there is no data in memory.	

:CALCulate:DATA:SIZE? <type>,<channel>,<year>,<month>,<day>,<hour>,<min>,<sec>

Parameter <type> = <CHARACTER PROGRAM DATA>

TTEXT Trace Text

RTEXT Recall Text

<channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16 CH1~CH16

ALL Total All Error

INSertion Total INS Error

OMISsion Total OMI Error

IOMission Total INS&OMI Error

<year>,<month>,<day>,<hour>,<min>,<sec> =

<DECIMAL NUMERIC PROGRAM DATA>

Response <NR1 NUMERIC RESPONSE DATA>

Returns the size of the data available at the specified point of time in bytes.

Function Queries the size of Trace Text data or Recall Text data available on a specified CH at a specified point of time. This command is used to allocate an input buffer for the controller to meet the data size.

:CALCulate:DATA:TEXT? <type>,<channel>,<year>,<month>,<day>,<hour>,<min>,<sec>

パラメータ <type> = <CHARACTER PROGRAM DATA>

TTEXT Trace Text

RTEXT Recall Text

<channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16 CH1~CH16

ALL Total All Error

INSertion Total INS Error

OMISsion Total OMI Error

IOMission Total INS&OMI Error

<year>,<month>,<day>,<hour>,<min>,<sec> =

<DECIMAL NUMERIC PROGRAM DATA>

Response <string> = <STRING RESPONSE DATA>

"Trace Text Data...."

Returns the tract text data available on a specified CH at a specified point of time.

"---None---"

There is no data for the specified point of time.

Function Queries the Trace Text data or Recall Text data available on a specified CH at a specified point of time.

Example Query the Trace Text Data available on CH1 at 1996-6-30 12:00:00.

:CALCulate:DATA:TEXT?△TTEXT,CH1,1996,6,30,12,0,0

Response : "**** Start Data ****,Start Time:96-02-16....;Slot3 :"

:CALCulate:DATA:STORe <string>

Parameter <string> = <STRING PROGRAM DATA>

"EALarm:LAST" Error/alarm measurement end data
 "EALarm:INtermediate" Error/alarm measurement intermediate data
 "DElay1" Delay (mode 1) measurement end data
 "DElay2" Delay (mode 2) measurement end data

Function Stores specified measurement data in the measurement result storage buffer. If measurement results are queried during iterative runs of measurement, the data could be switched in the middle as the current run of measurement is terminated and the next run of measurement begins before all the data is read. Use this command to buffer measurement data of interest and then read measurement results from the buffered data. The table below lists the kinds of data that are buffered.

Parameter	Buffered data
"EALarm:LAST"	Measurement end data and measurement stop time displayed onscreen when the command is received
"EALarm:INtermediate"	Measurement Intermediate data and the time at which that measurement data was created just before the reception of the command
"DElay1"	Delay measurement end data and measurement stop
"DElay2"	time just before the reception of the command

Example Query the ER (INS) end data on CH1 after the error/alarm measurement end data is buffered.

```
:CALCulate4:DATA:STORe "EALarm:LAST"
:CALCulate4:DATA:EALarm? 1,"LAST:ER:INS"
```

Note

1. Buffered data is retained until clear the buffer or the power is turned off.
2. Once data is buffered, it provides a Response to subsequent queries. If no data has been buffered, the current data displayed onscreen is returned. To clear the buffer, use the :CALCulate:DATA:CLEAr command.
3. Error/alarm current cannot be buffered. The current data displayed on screen is returned in response to queries of the current data.

:CALCulate:DATA:CLEAr <string>

Parameter <string> = <STRING PROGRAM DATA>

"EALarm:LAST" Error/alarm measurement end data
 "EALarm:INtermediate" Error/alarm measurement Intermediate data
 "DElay1" Delay (mode1) measurement end data
 "DElay2" Delay (mode2) measurement end data

Function Clears specified measurement data in the measurement result storage buffer.

6.4.4 OUTPut Subsystem

The OUTPut subsystem sets and queries SYNC Output settings.

Note:

Transmit commands by specifying :OUTPut4:... if the 200M 16CH ED is inserted in slot4.

Example: :OUTPut4:SYNC:SOURce CLOCK1

:OUTPut:SYNC:SOURce <source>

Parameter <sour> = <CHARACTER PROGRAM DATA>

CLOCK1 1/1 Clock Sync

CLOCK8 1/8 Clock Sync

PATtern Pattern Sync

Function Sets the SYNC Output output signal.

:OUTPut:SYNC:SOURce?

Response <sour> = <CHARACTER RESPONSE DATA>

Function Queries the SYNC Output output signal.

:OUTPut:SYNC:PATtern:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Function Sets the CH of the pattern with which SYNC output is synchronized when Pattern Sync has been selected as Sync Output.

:OUTPut:SYNC:PATtern:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

1 to 16 CH1 to CH16/Step:1

Function Queries the CH of the pattern with which SYNC output is synchronized when Pattern Sync has been selected as Sync Output.

:OUTPut:SYNC:PATtern:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>

PROGram, PRBS

Function Sets the type of Pattern with which SYNC output is synchronized.

:OUTPut:SYNC:PATtern:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the type of Pattern with which SYNC output is synchronized.

:OUTPut:SYNC:CLOCK:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1~CH16/Step:1

Function Sets the CH when 1/1 Clock Sync or 1/8 Clock Sync has been selected as Sync
 Output.

:OUTPut:SYNC:CLOCK:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>
 1 to 16 CH1~CH16/Step:1

Function Queries the CH when 1/1 Clock Sync or 1/8 Clock Sync has been selected as Sync
 Output.

6.4.5 INSTRUMENT Subsystem

The INSTRUMENT subsystem sets and queries groupings.

Note:

Transmit commands by specifying :INSTRUMENT4:... if the 200M 16CH ED is inserted in slot4.

Example: :INSTRUMENT4:COUPLE:CLOCK:GROUPING ON

:INSTRUMENT:COUPLE:PATTERN:SET <boolean>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	OFF or 0	Common slot3 and 4 settings OFF
	ON or 1	Common slot3 and 4 settings ON
Function	Sets the common setting ON/OFF status of pattern settings for two units.	

:INSTRUMENT:COUPLE:PATTERN:SET?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	OFF
	1	ON
Function	Queries the common setting ON/OFF status of pattern settings for two units.	

6.5 200M Clock Generator (MU163000A)

6.5.1 OUTPut Subsystem

The OUTPut subsystem sets and queries the transmitted signal setting of the 200M clock generator.

Note

Transmit commands by specifying :OUTPut1:... if the 200M clock generator is inserted in slot1.

Example: :OUTPut1:CLOCK:SOURce INTERNAL

:OUTPut:CLOCK:SOURce <source>

Parameter <source> = <CHARACTER PROGRAM DATA>
INTERNAL, EXTERNAL

Function Sets whether an external clock or internal clock is used for generating clock pulses.

:OUTPut:CLOCK:SOURce?

Response <source> = <CHARACTER RESPONSE DATA>

Function Queries whether an external clock or internal clock is used for generating clock pulses.

:OUTPut:CLOCK:FREQuency <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
10.0 to 1000.0 10.0 to 1000.0kHz/Step:0.1
1000 to 200000 1000 to 200000kHz/Step:1

Function Sets the clock frequency.

:OUTPut:CLOCK:FREQuency?

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the clock frequency.

:OUTPut:RClock:SElect <clock>

Parameter <clock> = <CHARACTER PROGRAM DATA>

 Internal Internal signal

 EXT10 EXT 10MHz

 EXT64 EXT 64kHz

Function Sets the reference signal for synchronizing internal and external signals when generating clock pulses.

:OUTPut:RClock:SElect?

Response <clock> = <CHARACTER RESPONSE DATA>

Function Queries the reference signal for synchronizing internal and external signals when generating clock pulses.

:OUTPut:JITter:EMODulation <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

 OFF or 0, ON or 1

Function Sets the ON/OFF status of the jitter modulation function.

:OUTPut:JITter:EMODulation?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

 0 OFF

 1 ON

Function Queries the ON/OFF status of the jitter modulation function.

:OUTPut:JITter:RANGe <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

 UI2 2UI

 UI50 50UI

Function Sets the jitter monitor range.

:OUTPut:JITter:RANGe?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the jitter monitor range.

6.5.2 ARM Subsystem

The ARM subsystem sets and queries the arming setting.

Note

Transmit commands by specifying :ARM1:... if the 200M clock generator is inserted in slot1.

Example: :ARM1:SOURce INTernal

:ARM:SOURce <source>

Parameter <source> = <CHARACTER PROGRAM DATA>
 INTernal, EXTernal

Function Sets an arming signal source.

:ARM:SOURce?

Parameter <source> = <CHARACTER PROGRAM DATA>

Function Queries the arming signal source.

:ARM:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Sets the ON/OFF status of INT Arming.

:ARM:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 OFF
 1 ON

Function Queries the ON/OFF status of INT Arming.

:ARM:START

Function Starts manual execute of Arming

6.5.3 CALCulate Subsystem

The CALCulate subsystem sets and queries the jitter monitor value and alarm monitor of the 200M clock generator.

Note

Transmit commands by specifying :CALCulate1:... if the 200M clock generator is inserted in slot1.

Example: :CALCulate1:JITTer:START

:CALCulate:DATA:MONitor? <string>

Parameter <string> = <STRING PROGRAM DATA>

See Table 6-4.

Response <string> = <STRING RESPONSE DATA>

Type	Format	Explanation
Form1	"X.XXX"	0~Max(UIpp)(Monitor Range:2UI)
	">X.XXX"	Over range(Monitor Range:2UI)
	"XX.XX"	0~Max(UIpp)(Monitor Range:50UI)
	">XX.XX"	Over range(Monitor Range:50UI)
	"-----"	There is no data for the query.
Form2	"Occur"	An alarm is present.
	(NL^END)	No alarm is present.
	"-----"	There is no data for the query.

Function Queries the data specified by the parameter.

Example Query the jitter monitor value.

:CALCulate1:DATA:MONitor?△"JITTer"

Response : "20.00"

Table 6-4 Monitor Items <string>

Item	<result>	Response format
Jitter Monitor	"JITTer"	Form1
PLL Unlock	"PLL"	Form2
External Clock Loss	"ECLoss"	Form2

6.6 Other Settings

6.6.1 DISPlay Subsystem

The DISPlay subsystem sets and queries the display operation settings.

:DISPlay:ACHannel <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Sets the ON/OFF status of the batch display of all the CHs for a 16CH unit.

:DISPlay:ACHannel?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 OFF

1 ON

Queries the ON/OFF status of the batch display of all the CHs for a 16CH unit.

:DISPlay:WINDow:OPEN <disp>

Parameter <disp> = <CHARACTER PROGRAM DATA>

SYSTem System subwindow

SETup Setup subwindow

TMENu Test Menu subwindow

RESult Result subwindow

ANALyze Analyze subwindow

CUSTomize Customize subwindow

Function Opens a subwindow.

:DISPlay:WINDow:OPEN?

Response <disp> = <CHARACTER RESPONSE DATA>

XXX,XXX,,, Returns open subwindow names, separated by a comma (,).

NONE No subwindow is open.

Function Queries open subwindow names.

:DISPlay:WINDow:CLOSE <disp>

Parameter <disp> = <CHARACTER PROGRAM DATA>
 SYSTem System subwindow
 SETup Setup subwindow
 TMENu Test Menu subwindow
 RESult Result subwindow
 ANALyze Analyze subwindow
 CUSTomize Customize subwindow

Function Closes a specified subwindow.

:DISPlay:SYSTem[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>
 SYSTem, OPTion

Function Switches the panel in the System subwindow.

:DISPlay:SYSTem[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>
 機能 Queries the panel in the System subwindow.

:DISPlay:SETup[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>
 SETup, FREQuency, CLOCk, DATA,
 PATTErn, AUX, TRIGger

Function Switches the panel in the Setup subwindow.

:DISPlay:SETup[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>
 Function Queries the panel in the Setup subwindow.

:DISPlay:SETup:CLOCK:CHANnel <unit>,<ch>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot3

4 Unit inserted in slot4

<ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Function Switches the CH of the group box on which the specified <unit> in the Setup:Clock I/F panel is displayed.

:DISPlay:SETup:CLOCK:CHANnel? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot3

4 Unit inserted in slot4

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the CH of the group box on which the specified <unit> in the Setup:Clock I/F panel is displayed.

:DISPlay:SETup:DATA:CHANnel <unit>,<ch>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

<ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

Function Switches the CH of the group box on which the specified <unit> in the Setup:Data I/F panel is displayed.

:DISPlay:SETup:DATA:CHANnel? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

Response <ch> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the CH of the group box on which the specified <unit> in the Setup:Data I/F panel is displayed.

:DISPlay:SETup:PATtern:PAGE <unit>,<page>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4
 <page> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 3 Page1 to Page3/Step:1

Function Switches the page of the group box on which the specified <unit> in the Setup:Pattern panel is displayed.

:DISPlay:SETup:PATtern:PAGE? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4
 Response <page> = <NR1 NUMERIC RESPONSE DATA>
 1 to 3 Page1 to Page3/Step:1

Function Queries the page of the group box on which the specified <unit> in the Setup:Pattern panel is displayed.

:DISPlay:SETup:PATtern:CHANnel <unit>,<ch>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4
 <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Switches the CH of the group box on which the specified <unit> in the Setup:Data I/F panel is displayed.

:DISPlay:SETup:PATtern:CHANnel? <unit>

 <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot3
 4 Unit inserted in slot4

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the CH of the group box on which the specified <unit> in the Setup:Data I/F panel is displayed.

:DISPlay:SETup:AUXiliary:PORT <unit>,<port>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

<ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 8 Port1 to Port8/Step:1

Function Switches the port of the group box on which the specified <unit> in the Setup:AUX I/F panel is displayed.

:DISPlay:SETup:AUXiliary:PORT? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

Response <port> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the port of the group box on which the specified <unit> in the Setup:AUX I/F panel is displayed..

:DISPlay:TEST[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>

MEASurement Measurement panel

EADDITION Error Addition panel

ANALyze Analyze panel

Function Switches the panel in the Test Menu subwindow.

:DISPlay:TEST[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>

Function Queries the panel in the Test Menu subwindow.

:DISPlay:TEST:EALarm:CHANnel <unit>,<ch>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4
 <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Switches the CH of the group box on which the specified <unit> in the Test Menu:Measurement (Test Item: Error/Alarm) panel is displayed.

:DISPlay:TEST:EALarm:CHANnel? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the CH of the group box on which the specified <unit> in the Test Menu:Measurement (Test Item: Error/Alarm) panel is displayed.

:DISPlay:TEST:EADDITION:CHANnel <unit>,<ch>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4
 <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Switches the CH of the group box on which the specified <unit> in the Test Menu:Error Addition panel is displayed.

:DISPlay:TEST:EADDITION:CHANnel? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the CH of the group box on which the specified <unit> in the Test Menu:Error Addition panel is displayed.

:DISPlay:RESult:TIME <time>

Parameter <time> = <CHARACTER PROGRAM DATA>
 ELAPsed Elapsed time display
 TIMed Timed display
 DTIMe Date & Time display
 START Start display

Function Sets the Result subwindow measurement time display mode.

:DISPlay:RESult:TIME?

Response <time> = <CHARACTER RESPONSE DATA>

Function Result Queries the Result subwindow measurement time display mode.

:DISPlay:RESult:EALarm[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>
 ALL, NORMAl, ZOOM, MONitor

Function Switches the Result subwindow (Error/Alarm) panel.

:DISPlay:RESult:EALarm[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>

Function Queries the Result subwindow (Error/Alarm) panel.

:DISPlay:RESult:EALarm:MODE <mode>

 <mode> = <CHARACTER PROGRAM DATA>
 CURRent, LAST

Function Sets the Result subwindow (Error/Alarm) display mode.

:DISPlay:RESult:EALarm:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the Result subwindow (Error/Alarm) display mode.

:DISPlay:RESult:EALarm:TYPE <type>

Parameter <type> = <CHARACTER PROC RAM DATA>

ER	Error Rate
EC	Error Count
EI	EI
EFI	%EFI
TEI	Threshold EI
TEFI	Threshold %EFI
PERFormance	Performance

Function Sets the error measurement item to be displayed on the Result:ALL panel.

:DISPlay:RESult:EALarm:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>

Function Queries the error measurement item to be displayed on the Result:ALL panel.

:DISPlay:RESult:EALarm:PITem <pitem>

Parameter <pitem> = <CHARACTER PROGRAM DATA>

ES	%ES		ES2	ES
EFS	%EFS		EFS2	EFS
SES	%SES		SES2	SES
DM	%DM		DM2	DM
US	%US		US2	US
			EC	EC

Function Sets the performance measurement item to be displayed when Performance has been selected as the Result:ALL panel display item.

:DISPlay:RESult:EALarm:PITem?

Response <pitem> = <CHARACTER PROGRAM DATA>

Function Queries the performance measurement item to be displayed when Performance has been selected as the Result:ALL panel display item..

:DISPlay:RESult:EALarm:TITem <titem>

Parameter <titem> = <CHARACTER PROGRAM DATA>

E_3	>1.0E-3		E_7	>1.0E-7
E_4	>1.0E-4		E_8	>1.0E-8
E_5	>1.0E-5		UE_8	=<1.0E-8
E_6	>1.0E-6			

Function Sets the item to be displayed when Threshold EI or Threshold %EFI has been selected as the Result:ALL panel display item.

:DISPlay:RESult:EALarm:TITem?

Response <titem> = <CHARACTER PROGRAM DATA>

Function Queries the item to be displayed when Threshold EI or Threshold %EFI has been selected as the Result:ALL panel display item.

:DISPlay:RESult:EALarm:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3	Unit inserted in slot 3
4	Unit inserted in slot 4

Function Sets the unit to be displayed on the Result:ALL or Normal panel.

:DISPlay:RESult:EALarm:UNIT?

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the unit to be displayed on the Result:ALL or Normal panel.

:DISPlay:RESult:EALarm:CHANnel <channel>

Parameter <channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16	CH1 to CH16
ALL	Total ALL Error
INSertion	Total INS Error
OMISSion	Total OMI Error
IOMission	Total INS&OMI Error

Function Sets the CH to be displayed on the Result:Normal panel.

:DISPlay:RESult:EALarm:CHANnel?

Response <channel> = <CHARACTER RESPONSE DATA>

Function Queries the CH to be displayed on the Result:Normal panel.

:DISPlay:RESult:EALarm:ZOOM1:SET <boolean>

:DISPlay:RESult:EALarm:ZOOM2:SET <boolean>

:DISPlay:RESult:EALarm:ZOOM3:SET <boolean>

:DISPlay:RESult:EALarm:ZOOM4:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Sets the display ON/OFF status of the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:SET?

:DISPlay:RESult:EALarm:ZOOM2:SET?

:DISPlay:RESult:EALarm:ZOOM3:SET?

:DISPlay:RESult:EALarm:ZOOM4:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 OFF
 1 ON

Function Result:Zoom Queries the display ON/OFF status of the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:UNIT <unit>

:DISPlay:RESult:EALarm:ZOOM2:UNIT <unit>

:DISPlay:RESult:EALarm:ZOOM3:UNIT <unit>

:DISPlay:RESult:EALarm:ZOOM4:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4

Function Sets the unit to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:UNIT?

:DISPlay:RESult:EALarm:ZOOM2:UNIT?

:DISPlay:RESult:EALarm:ZOOM3:UNIT?

:DISPlay:RESult:EALarm:ZOOM4:UNIT?

Response <unit> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the unit to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:CHANnel <ch>

:DISPlay:RESult:EALarm:ZOOM2:CHANnel <ch>

:DISPlay:RESult:EALarm:ZOOM3:CHANnel <ch>

:DISPlay:RESult:EALarm:ZOOM4:CHANnel <ch>

Parameter <channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16	CH1 to CH16
ALL	Total ALL Error
INSertion	Total INS Error
OMISsion	Total OMI Error
IOMission	Total INS&OMI Error

Function Sets the CH to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:CHANnel?

:DISPlay:RESult:EALarm:ZOOM2:CHANnel?

:DISPlay:RESult:EALarm:ZOOM3:CHANnel?

:DISPlay:RESult:EALarm:ZOOM4:CHANnel?

Response <channel> = <CHARACTER RESPONSE DATA>

Function Queries the CH to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:ITEM <item>

:DISPlay:RESult:EALarm:ZOOM2:ITEM <item>

:DISPlay:RESult:EALarm:ZOOM3:ITEM <item>

:DISPlay:RESult:EALarm:ZOOM4:ITEM <item>

Parameter <item> = <CHARACTER PROGRAM DATA>

ER	Error Rate
EC	Error Count
EI	EI
EFI	%EFI
PSLoss	Pattern Sync Loss
CLOSs	Clock Loss
PFail	Power Fail
TEI	Threshold EI
TEFI	Threshold %EFI
PERFormance	Performance
FREQuency	Frequency

Function Sets the item to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:ITEM?

:DISPlay:RESult:EALarm:ZOOM2:ITEM?

:DISPlay:RESult:EALarm:ZOOM3:ITEM?

:DISPlay:RESult:EALarm:ZOOM4:ITEM?

Response <item> = <CHARACTER PROGRAM DATA>

Function Queries the item to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:PITem <pittem>

:DISPlay:RESult:EALarm:ZOOM2:PITem <pittem>

:DISPlay:RESult:EALarm:ZOOM3:PITem <pittem>

:DISPlay:RESult:EALarm:ZOOM4:PITem <pittem>

Parameter <pittem> = <CHARACTER PROGRAM DATA>

ES	%ES		ES2	ES
EFS	%EFS		EFS2	EFS
SES	%SES		SES2	SES
DM	%DM		DM2	DM
US	%US		US2	US
			EC	EC

Function Sets the performance measurement item to be displayed when Performance has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:ZOOM1:PITem?

:DISPlay:RESult:EALarm:ZOOM2:PITem?

:DISPlay:RESult:EALarm:ZOOM3:PITem?

:DISPlay:RESult:EALarm:ZOOM4:PITem?

Response <pittem> = <CHARACTER RESPONSE DATA>

Function Queries the performance measurement item to be displayed when Performance has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:ZOOM1:TITtem <titem>

:DISPlay:RESult:EALarm:ZOOM2:TITtem <titem>

:DISPlay:RESult:EALarm:ZOOM3:TITtem <titem>

:DISPlay:RESult:EALarm:ZOOM4:TITtem <titem>

Parameter <titem> = <CHARACTER PROGRAM DATA>

E_3	>1.0E-3		E_7	>1.0E-7
E_4	>1.0E-4		E_8	>1.0E-8
E_5	>1.0E-5		UE_8	=<1.0E-8
E_6	>1.0E-6			

Function Sets the performance measurement item to be displayed when Threshold EI or Threshold %EFI has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:ZOOM1:TITtem?

:DISPlay:RESult:EALarm:ZOOM2:TITtem?

:DISPlay:RESult:EALarm:ZOOM3:TITtem?

:DISPlay:RESult:EALarm:ZOOM4:TITtem?

Response <titem> = <CHARACTER RESPONSE DATA>

Function Queries the performance measurement item to be displayed when Threshold EI or Threshold %EFI has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:HRESet

Function Resets the Result:Monitor panel history lamp.

:DISPlay:RESult:EMARgin:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

RESult Result

RDiagram Recall Diagram

Function Switchs the display of Result Subwindow(Eye Margin measurement).

:DISPlay:RESult:EMARgin:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the display of Result Sub window(Eye Margin measurement).

:DISPlay:RESult:EMARgin:ERATe <erate>,<boolean>

Parameter <erate> = <CHARACTER PROGRAM DATA>

E_3, E_4, E_5, E_6, E_7, E_8, E_9

Correspond to 1E-3, 1E-4,..., and 1E-9, respectively.

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the Result subwindow (eye margin measurement) display error rate.
The subject of this commad is depend of MODE selection.
When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram
(Recall Diagram).

:DISPlay:RESult:EMARgin:ERATe?

Response <erate> = <CHARACTER PROGRAM DATA>

E_X,E_X,.. Returns the error rates that are displayed, separated
by a comma (,).

NONE Noting is displayed.

Function Queries the Result subwindow (eye margin measurement) display error rate.
The subject of this commad is depend of MODE selection.
When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram
(Recall Diagram).

:DISPlay:RESult:EMARgin:SCALE:AScale

Function Execute the automatic setup of the scale in the Result subwindow (eye margin
measurement) display area.
The subject of this commad is depend of MODE selection.
When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram
(Recall Diagram).

:DISPlay:RESult:EMARgin:SCALE:VOLTage <min>,<step>

Parameter <min> = <DECIMAL NUMERIC PROGRAM DATA>
 -5.000 to 4.950 -5.000 to 4.950 V/Step:0.005
 <step> = <DECIMAL NUMERIC PROGRAM DATA>
 0.005 to 1.000 0.005 to 1.000 V/Step:0.005

Function Sets the minimum value of the vertical scale (Voltage) in the Result subwindow (eye margin measurement) graph.
 The subject of this command is depend of MODE selection.
 When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
 When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:SCALE:VOLTage?

Response <min>,<step> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the minimum value of the vertical scale (Voltage) in the Result subwindow (eye margin measurement) graph.
 The subject of this command is depend of MODE selection.
 When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
 When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:SCALE:PHASe <min>,<step>

Parameter <min> = <DECIMAL NUMERIC PROGRAM DATA>
 -5.0 to 4.0 -5.0 to 4.0 ns/Step:0.1
 <step> = <DECIMAL NUMERIC PROGRAM DATA>
 0.1 to 1.0 0.1 to 1.0 ns/Step:0.1

Function Sets the minimum value of the horizontal scale (Phase) in the Result subwindow (eye margin measurement) graph.
 The subject of this command is depend of MODE selection.
 When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
 When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:SCALE:PHASe?

Response <min>,<step> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the minimum value and step of the horizontal scale in the Result subwindow (eye margin measurement) graph.
 The subject of this command is depend of MODE selection.
 When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
 When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:SET <marker>,<boolean>

Parameter <marker> = <CHARACTER PROGRAM DATA>
 BASE Base marker
 REFerence Reference marker
 <boolean> = <BOOLEAN PROGRAM DATA>
 OFF or 0, ON or 1

Function Sets the ON/OFF status of the Result subwindow (eye margin measurement) marker.
 The subject of this commad is depend of MODE selection.
 When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
 When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:SET? <marker>

Parameter <marker> = <CHARACTER PROGRAM DATA>
 BASE Base marker
 REFerence Reference marker

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
 0 OFF
 1 ON

Function Queries the ON/OFF status of the Result subwindow (eye margin measurement) marker.
 The subject of this commad is depend of MODE selection.
 When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
 When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
 FREE Allows the marker to move to any position.
 FIX Allows the marker to move along the measurement line.

Function Sets the movement mode of the Result subwindow (eye margin measurement) marker.
 The subject of this commad is depend of MODE selection.
 When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).
 When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the movement mode of the Result subwindow (eye margin measurement) marker.

The subject of this command is depend of MODE selection.

When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).

When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:ERATe <erate>

Parameter <erate> = <CHARACTER PROGRAM DATA>

E_3, E_4, E_5, E_6, E_7, E_8, E_9

Correspond to 1E-3, 1E-4,..., and 1E-9, respectively.

Function Sets along which error rate line displayed the Result subwindow (eye margin measurement) marker moves.

The subject of this command is depend of MODE selection.

When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).

When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:ERATe?

Response <erate> = <CHARACTER RESPONSE DATA>

Function Queries along which error rate line displayed the Result subwindow (eye margin measurement) marker moves.

The subject of this command is depend of MODE selection.

When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).

When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:MOVE <marker>,<mode>

Parameter <marker> = <CHARACTER PROGRAM DATA>

BASE Base marker

REFeRence Reference marker

<mode> = <CHARACTER PROGRAM DATA>

LEFT Moves to left or counterclockwise.

RIGHT Moves to right or clockwise.

UP Moves up or clockwise.

DOWN Moves down or counterclockwise.

Function Moves the Result subwindow (eye margin measurement) marker.

The subject of this command is depend of MODE selection.

When Result is selected, subject subwindow is the Eye Margin/Diagram(Result).

When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).

:DISPlay:RESult:EMARgin:MARKer:POSition? <marker>

Parameter	<marker> = <CHARACTER PROGRAM DATA>	
	BASE	Base marker
	REFerence	Reference marker
Response	<phase>, <voltage>	
	<phase> = <NR2 NUMERIC RESPONSE DATA>	
	-X.X to X.X Horizontal (Phase) value	
Function	<voltage> = <NR2 NUMERIC RESPONSE DATA>	
	-X.XXX to X.XXX Vertical (Threshold voltage) value	
	Queries the Result subwindow (eye margin measurement) marker. The absolute position of the base marker, or the position of a reference marker relative to the base marker, is returned. The subject of this commad is depend of MODE selection. When Result is selected, subject subwindow is the Eye Margin/Diagram(Result). When Recall Diagram is selected, subject subwindow is the Eye Margin/Diagram (Recall Diagram).	

:DISPlay:CUSTomize:SETup1:UNIT <unit>

:DISPlay:CUSTomize:SETup2:UNIT <unit>

:DISPlay:CUSTomize:SETup3:UNIT <unit>

:DISPlay:CUSTomize:SETup4:UNIT <unit>

:DISPlay:CUSTomize:SETup5:UNIT <unit>

:DISPlay:CUSTomize:SETup6:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

0 OFF

1 Unit inserted in slot 1

3 Unit inserted in slot 3

4 Unit inserted in slot 4

Function Sets the units to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:UNIT?

:DISPlay:CUSTomize:SETup2:UNIT?

:DISPlay:CUSTomize:SETup3:UNIT?

:DISPlay:CUSTomize:SETup4:UNIT?

:DISPlay:CUSTomize:SETup5:UNIT?

:DISPlay:CUSTomize:SETup6:UNIT?

Response <unit> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the units to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:CHANnel <ch>

:DISPlay:CUSTomize:SETup2:CHANnel <ch>

:DISPlay:CUSTomize:SETup3:CHANnel <ch>

:DISPlay:CUSTomize:SETup4:CHANnel <ch>

:DISPlay:CUSTomize:SETup5:CHANnel <ch>

:DISPlay:CUSTomize:SETup6:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Sets the CHs to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:CHANnel?

:DISPlay:CUSTomize:SETup2:CHANnel?

:DISPlay:CUSTomize:SETup3:CHANnel?

:DISPlay:CUSTomize:SETup4:CHANnel?

:DISPlay:CUSTomize:SETup5:CHANnel?

:DISPlay:CUSTomize:SETup6:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the CHs to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:ITEM <string>
:DISPlay:CUSTomize:SETup2:ITEM <string>
:DISPlay:CUSTomize:SETup3:ITEM <string>
:DISPlay:CUSTomize:SETup4:ITEM <string>
:DISPlay:CUSTomize:SETup5:ITEM <string>
:DISPlay:CUSTomize:SETup6:ITEM <string>

Parameter <string> = <STRING PROGRAM DATA>

"FREQuency"	Frequency
"CLOCK:AMPLitude"	Clock/Amplitude
"CLOCK:OFFSet"	Clock/Offset
"CLOCK:DELAy"	Clock/Delay
"CLOCK:POLarity"	Clock/Polarity
"CLOCK:AMPLitude"	Clock/Amplitude
"DATA:AMPLitude"	Data/Amplitude
"DATA:OFFSet"	Data/Offset
"DATA:SKEW"	Data/Skew
"DATA:THReshold"	Data/Threshold
"PATtern:LOGic"	Pattern/Logic
"PATtern:MRATio"	Pattern/Mark Ratio
"BTDelay"	Burst Trigger Delay

Function Sets the items to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:ITEM?
:DISPlay:CUSTomize:SETup2:ITEM?
:DISPlay:CUSTomize:SETup3:ITEM?
:DISPlay:CUSTomize:SETup4:ITEM?
:DISPlay:CUSTomize:SETup5:ITEM?
:DISPlay:CUSTomize:SETup6:ITEM?

Response <string> = <STRING RESPONSE DATA>

Function Queries the items to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:PATtern:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

0	OFF
3	Unit inserted in slot 3
4	Unit inserted in slot 4.

Function Sets the unit to be displayed in the Customize subwindow Pattern group box.

:DISPlay:CUSTomize:PATtern:UNIT?

Response <unit> = <CHARACTER RESPONSE DATA>

Function Queries the unit to be displayed in the Customize subwindow Pattern group box.

:DISPlay:CUSTomize:PATtern:CHANnel <ch>

Parameter <ch> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 16 CH1 to CH16/Step:1

Function Sets the CH to be displayed in the Customize subwindow Pattern group box.

:DISPlay:CUSTomize:PATtern:CHANnel?

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the CH to be displayed in the Customize subwindow Pattern group box.

:DISPlay:CUSTomize:PATtern:OFFSet <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 0 to MAX 0 to MAX bits/Step:8
 MAX is "Length setting of the pattern-8" when editing pattern is
 PRGM.
 MAX is "Length setting of the block-8" when editing pattern is
 MIX or PON.

Function Specifies the bit position of the pattern to be edited as a byte boundary.

:DISPlay:CUSTomize:PATtern:OFFSet?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit position of the pattern to be edited.

:DISPlay:CUSTomize:PATtern:BLOCK <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 32 #1 to #32/Step:1

Function Sets the block of the pattern to be edited on the Customize panel.

:DISPlay:CUSTomize:PATtern:BLOCK?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the block of the pattern to be edited on the Customize panel.

:DISPlay:CUSTomize:RESult:UNIT <unit>

Parameter <unit> = <CHARACTER PROGRAM DATA>
 0 OFF
 3 Unit inserted in slot 3
 4 Unit inserted in slot 4

Function Sets the unit to be displayed in the Customize subwindow Result group box.

:DISPlay:CUSTomize:RESult:UNIT?

Response <unit> = <CHARACTER RESPONSE DATA>

Function Queries the unit to be displayed in the Customize subwindow Result group box.

:DISPlay:CUSTomize:RESult:CHANnel <channel>

Parameter <channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16	CH1 to CH16
ALL	Total ALL Error
INSertion	Total INS Error
OMISsion	Total OMI Error
IOMission	Total INS&OMI Error

Function Sets the CH to be displayed in the Customize subwindow Result group box.

:DISPlay:CUSTomize:RESult:CHANnel?

Response <channel> = <CHARACTER RESPONSE DATA>

Function Queries the CH to be displayed in the Customize subwindow Result group box.

:DISPlay:CUSTomize:RESult:ITEM <item>

Parameter <item> = <CHARACTER PROGRAM DATA>

ER	Error Rate
EC	Error Count

Function Sets the item to be displayed in the Customize subwindow Result group box.

:DISPlay:CUSTomize:RESult:ITEM?

Response <item> = <CHARACTER RESPONSE DATA>

Function Queries the item to be displayed in the Customize subwindow Result group box.

:DISPlay:ANALyze:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

TTEXT	Trace Text
RTEXT	Recall Text

Function Switches the Analyze function (screen).

:DISPlay:ANALyze:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

Function Queries the Analyze function (screen).

:DISPlay:ANALyze:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3	Unit inserted in slot3
4	Unit inserted in slot4

Function Sets the unit to be displayed in the Analyze subwindow.
 The subject of this commad is depend of MODE selection.
 When Trace Text is selected, subject subwindow is the Analyze(Trace Text).
 When Recall Text is selected, subject subwindow is the Analyze(Recall Text).

:DISPlay:ANALyze:UNIT?

Response <unit> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the unit to be displayed in the Analyze subwindow.
 The subject of this commad is depend of MODE selection.
 When Trace Text is selected, subject subwindow is the Analyze(Trace Text).
 When Recall Text is selected, subject subwindow is the Analyze(Recall Text).

:DISPlay:ANALyze:CHANnel <channel>

Parameter <channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16	CH1 to CH16
ALL	Total ALL Error
INSertion	Total INS Error(INS/OMI)
OMISSion	Total OMI Error(INS/OMI)
IOMission	Total INS&OMI Error

Function Sets the CH to be displayed in the Analyze subwindow.
 The subject of this commad is depend of MODE selection.
 When Trace Text is selected, subject subwindow is the Analyze(Trace Text).
 When Recall Text is selected, subject subwindow is the Analyze(Recall Text).

:DISPlay:ANALyze:CHANnel?

Response <channel> = <CHARACTER RESPONSE DATA>

Function Queries the CH to be displayed in the Analyze subwindow.
 The subject of this commad is depend of MODE selection.
 When Trace Text is selected, subject subwindow is the Analyze(Trace Text).
 When Recall Text is selected, subject subwindow is the Analyze(Recall Text).

:DISPlay:ANALyze:TEXT:SCRoll <scroll>

Parameter <scroll> = <CHARACTER PROGRAM DATA>

UP	Scrolls up.
DOWN	Scrolls down.
TOP	Scrolls to the beginning of data.
BOTTOM	Scrolls to the end of data.

Function Scrolls the Analyze subwindow (Mode:Trace Text) screen.
 The subject of this commad is depend of MODE selection.
 When Trace Text is selected, subject subwindow is the Analyze(Trace Text).
 When Recall Text is selected, subject subwindow is the Analyze(Recall Text).

```
:DISPlay:ANALyze:TEXT:TIME <year>,<month>,<day>,<hour>,<min>,<sec>
```

Parameter	<year>,<month>,<day>,<hour>,<min>,<sec> = <DECIMAL NUMERIC PROGRAM DATA>
-----------	---

Function Sets the display position of tract text data in time terms.
The subject of this command is depend of MODE selection.
When Trace Text is selected, subject subwindow is the Analyze(Trace Text).
When Recall Text is selected, subject subwindow is the Analyze(Recall Text).

```
:DISPlay:ANALyze:TEXT:TIME?
```

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function	Queries the display position of tract text data. The subject of this command is depend of MODE selection. When Trace Text is selected, subject subwindow is the Analyze(Trace Text). When Recall Text is selected, subject subwindow is the Analyze(Recall Text).
----------	---

6.6.2 SYSTem Subsystem

The SYSTem subsystem sets and queries the beeper, date, and other settings.

:SYSTem:BEEPer:ERROr:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets the ON/OFF status of the beeper on occurrence of errors.

:SYSTem:BEEPer:ERROr:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 OFF
1 ON

Function Queries the ON/OFF status of the beeper on occurrence of errors.

:SYSTem:BEEPer:ERROr:CHANnel <unit>,<ch>,<boolean>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
3 Unit inserted in slot 3
4 Unit inserted in slot 4
<ch> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 CH1 to CH16/Step:1
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets the ON/OFF status of the error beeper for a specified unit on a specified CH.

:SYSTem:BEEPer:ERROr:CHANnel? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
3 Unit inserted in slot 3
4 Unit inserted in slot 4

Response <ch> = <NR1 NUMERIC RESPONSE DATA>
X,X,.. Returns the CHs having their error beeper set ON, separated
by a comma (.).
0 All CHs have their error beeper set OFF.

Function Queries the CHs that have their error beeper set ON.

:SYSTem:BEEPer:ALARm:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of the alarm beeper on occurrence of alarms.

:SYSTem:BEEPer:ALARm:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 OFF

1 ON

Function Queries the ON/OFF status of the alarm beeper on occurrence of alarms.

:SYSTem:BEEPer:ALARm:CHANnel <unit>,<ch>,<boolean>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

<ch> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16 CH1 to CH16/Step:1

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the ON/OFF status of the alarm beeper for a specified unit on a specified CH.

:SYSTem:BEEPer:ALARm:CHANnel? <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

Response <ch> = <NR1 NUMERIC RESPONSE DATA>

X,X,... Returns the CHs having their beeper set ON, separated by a
comma (,).

0 All CHs have their beeper set OFF.

Function Queries the CHs that have their alarm beeper set ON.

:SYSTem:BEEPer:SYSTem:SET <boolean>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA> OFF or 0, ON or 1
-----------	---

Function Sets the ON/OFF status of the system error beeper.

:SYSTem:BEEPer:SYSTem:SET?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	OFF
	1	ON

Function	Queries the ON/OFF status of the system error beeper.
----------	---

```
:SYSTem:BEEPer:SYSTem:TYPE <type>,<boolean>
```

Parameter	<type> = <CHARACTER PROGRAM DATA>
	PUNLock PLL Unlock(Slot1)
	ECLoss EXT Clock Loss(Slot1)
	PCLoss3 PPG Clock Loss(Slot3)
	FEMpty4 FIFO Empty(Slot4)
	FFULl4 FIFO FULL(Slot4)
	<boolean> = <BOOLEAN PROGRAM DATA>
	OFF or 0, ON or 1

Function Sets the ON/OFF status of items that are signaled by the system error beeper.

```
:SYSTem:BEEPer:SYSTem:TYPE?
```

Response	<type> = <CHARACTER RESPONSE DATA>
XXX,XXX,..	Returns the errors having the system error beeper set ON, separated by a comma (.).
NONE	No errors have the system error beeper set ON.

Function	Queries the ON/OFF status of items that are signaled by the system error beeper.
----------	--

:SYSTem:MODE?

Parameter <mode> = <CHARACTER RESPONSE DATA>
 NORM Normal Mode
 PFA Measurement Power Fail Mode

Function Queries the mode of system operation.

:SYSTem:DATE <year>,<month>,<day>

Parameter <year>,<month>,<day> = <DECIMAL NUMERIC PROGRAM DATA>
 <year>= 1996 to 2037/Step:1
 <month>= 1 to 12/Step:1
 <day>= 1 to 31/Step:1

Function Sets the current date.

:SYSTem:DATE?

Response <year>,<month>,<day> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the current date.

:SYSTem:TIME <hour>,<min>,<sec>

Parameter <hour>,<min>,<sec> = <DECIMAL NUMERIC PROGRAM DATA>
 <hour>= 0 to 23/Step:1
 <min>= 0 to 59/Step:1
 <sec>= 0 to 59/Step:1

Function Sets the current time.

:SYSTem:TIME?

Response <hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the current time.

:SYSTem:BSIZe? <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
 INPut Input buffer size
 OUTPut Output buffer size
 Response <size> = <NR1 NUMERIC RESPONSE DATA>
 Returns the I/O buffer size in bytes.
 Function Queries the size of the instrument's I/O buffers.

:SYSTem:ERRor?

Response <error/event_number>,<error/event_description>
 <error/event_number> = <NR1 NUMERIC RESPONSE DATA>
 -32768 to 32767
 <error/event_description> = <STRING RESPONSE DATA>
 Function Queries the error status. See Appendix B for a detailed definition of errors.

:SYSTem:VERsion?

Response <version> = <NR2 NUMERIC RESPONSE DATA>
 YYYY.V YYYY:year
 V:revision number
 Function Queries the version to which this instrument conforms. This instrument will
 return <1995.0>.

:SYSTem:ORGanization:HRADware?

Response <slot1>, <slot2>, <slot3>, <slot4>
 <slotX> = <unit_name>, <version>, <option>
 <unit_name> = <STRING RESPONSE DATA> Unit name
 "200M Clock" 200M Clock Generator
 "200M 16CH PPG" 200M 16CH PPG Unit
 "200M 16CH ED" 200M 16CH ED Unit
 "... " No unit is inserted in the slot.
 <version> = <NR1 NUMERIC RESPONSE DATA>
 1 to XX Unit version
 0 No unit is inserted in the slot.
 <option> = <STRING RESPONSE DATA>
 "OPTXX:OPTXX.." Returns unit options, separated by a comma
 (.).
 OPTXX = OPT01, OPT02,.. (XX: Two digits)
 "NONE" No option is installed.
 "... " No unit is inserted in the slot.

Function Queries the instrument's system configuration. Use command commands to query the mainframe version and options.

:SYSTem:ORGanization:SOFTware?

Response <NR2 NUMERIC RESPONSE DATA>
 X.X Version X.X

Function Queries the instrument's software system version.

:SYSTem:MMEMory:RECall <type>,<file_name>,<unit>,<ch>

Parameter <type> = <STRING PROGRAM DATA>

"MAIN"	Main Frame Setup
"SLOT1"	Slot1 Setup
"PPG"	PPG Setup
"ED"	ED Setup
"PATtern"	Pattern Setup
"TTExT:BINary"	Analyze Text(Binary format)
"EYE"	Eye Diagram Data

 <file_name> = <STRING PROGRAM DATA>

 "<drv>:¥[<dir>]<file>"

 <drv>= A, C, ... (Available drives)

 <dir>= <dir1>¥<dir2>¥...(Omitted for a root directory)

 <file> = filename

 <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3	Unit inserted in slot 3
4	Unit inserted in slot 4
0	Any other type of file

 If the type of file to read is PPG/ED Setup or Pattern Data, specify the unit to read to.

 <ch> = <DECIMAL NUMERIC PROGRAM DATA>

0	Normal (reading a file having the same number of channels)
1 to 16	CH1 to CH16/Step:1

 Specifies the CH to read from or read to when reading a file containing the status of a unit having a different number of channels).

Function Reads a specified file to a specified unit.

:SYSTem:MMEMory:STORe <type>,<file_name>,<unit>

Parameter <type> = <STRING PROGRAM DATA>

"MAIN"	Main Frame Setup
"SLOT1"	Slot1 Setup
"PPG"	PPG Setup
"ED"	ED Setup
"PATtern"	Pattern Setup
"TTExt:BINary"	Analyze Text(Binary format)
"TTExt:TEXT"	Analyze Text(Text format)
"EYE"	Eye Diagram Data

<file_name> = <STRING PROGRAM DATA>

"<drv>:¥[<dir>]<file>"

<drv>= A, C, ... (Available drives)

<dir>= <dir1>¥<dir2>¥... (Omitted for a root directory)

<file> = filename (extension not required)

<unit> = <DECIMAL NUMERIC PROGRAM DATA>

3	Unit inserted in slot 3
4	Unit inserted in slot 4
0	Source unit setting is not required.

Specify the unit (slot) to save.

Function Saves specified data from a specified unit to a file.

:SYSTem:MMEMory:QRECall <file_name>

Parameter <file_name> = <STRING PROGRAM DATA>

"<drv>:¥[<dir>]<file>"

<drv>= A, C, ... (Available drives)

<dir>= <dir1>¥<dir2>¥... (Omitted for a root directory)

<file> = filename (use the extension .ENV)

Function Quick-opens.

:SYSTem:MMEMory:QSTore <file_name>,<comment>

Parameter <file_name> = <STRING PROGRAM DATA>

"<drv>:¥[<dir>]<file>"

<drv>= A, C, ... (Available drives)

<dir>= <dir1>¥<dir2>¥... (Omitted for a root directory)

<file> = Filename(extension not required)

<comment> = <STRING PROGRAM DATA>

"XXXX..." Up to 60 characters

Specify a file comment.

Function Quick-saves.

:SYSTem:MMEMory:CATalog? <drv_dir>,<type>

Parameter	<drv_dir> = <STRING PROGRAM DATA> "<drv>:¥[<dir>]" <drv>= A, C, ... (Available drives) <dir>= <dir1>¥<dir2>¥... (Omitted for a root directory) <type> = <STRING PROGRAM DATA> "MAIN" Main Frame Setup "SLOT1" Slot1 Setup "PPG" PPG Setup "ED" ED Setup "PATTErn" Pattern Setup "TTExt:BiNary" Analyze Text(Binary format) "TTExt:TEXT" Analyze Text(Text format) "EYE" Eye Diagram Data "ALL" All data
Response	<use_byte>,<free_byte>,<current_dir>,<current_file>,<file_entry> <use_byte> = <NT1 NUMERIC RESPONSE DATA> Space used (bytes) <free_byte> = <NR1 NUMERIC RESPONSE DATA> Available space (bytes) <current_dir> = <STRING RESPONSE DATA> Current directory <current_file> = <NR1 NUMERIC RESPONSE DATA> Number of files that match the <type> specification existing in the current directory, including subdirectories if "<ALL>" has been specified as <type>. <file_entry> = <STRING RESPONSE DATA> Returns, in the following format, files or directories that match <type>, or "" if no files match. "<file_name>,<file_size>,<date_time>,..." <file_name> Filename or subdirectory name XXXX.XXX Filename <XXXX> : Directory name <file_size> File size XXX file size (bytes) <date_time> Date of file creation <year>.<month>.<day>△<hour>:<min>:<sec>
Function	Queries HDD or FDD file information. 0, 0, "...", 0, "" is returned when disk information cannot be read.

:SYSTem:MEMory:INITialize

Function Resets the instrument settings to their factory defaults.

:SYSTem:PRINt:COpy

Function Copies the contents currently displayed on the screen.

:SYSTem:PRINt:TEXT <string>

Parameter <string> = <STRING PROGRAM DATA>

"character string" Up to 40 characters

Function Outputs the character string specified by the parameter to the printer.

:SYSTem:PRINt:EALarm <item>

Parameter <item> = <STRING PROGRAM DATA>

"XXX:YYY:..." Sets output items, separated by a colon (:).

Output items

ERRor Error

THReshold Threshold EI/%EFI

PCOunt Performance Count

PPERcent Performance %

ALARm Alarm Interval

FREQuency Clock Frequency Data

Function Prints the specified items of immediate error/alarm measurement data.

:SYSTem:PRINT:TTEXT:TIME <unit>,<channel>,

<y1>,<m1>,<d1>,<h1>,<n1>,<s1>,<y2>,<m2>,<d2>,<h2>,<n2>,<s2>

:SYSTem:PRINT:TTEXT:ALL <unit>,<channel>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

<channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16 CH1 to CH16

ALL Total All Error

INSertion Total INS Error

OMISsion Total OMI Error

IOMission Total INS&OMI Error

<y1>,<m1>,<d1>,<h1>,<n1>,<s1> = <DECIMAL NUMERIC PROGRAM DATA>

Print start time (year/month/date/hour/minute/second)

<y2>,<m2>,<d2>,<h2>,<n2>,<s2> = <DECIMAL NUMERIC PROGRAM DATA>

Print end time (year/month/date/hour/minute/second)

Function Prints the trace text data for a specified unit on a specified CH in effect at a specified point of time (:...:TIME command).

Prints all the trace text data for a specified unit on a specified CH (:...:ALL command).

:SYSTem:PRINT:RTEXT:TIME <unit>,<channel>,

<y1>,<m1>,<d1>,<h1>,<n1>,<s1>,<y2>,<m2>,<d2>,<h2>,<n2>,<s2>

:SYSTem:PRINT:RTEXT:ALL <unit>,<channel>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot 3

4 Unit inserted in slot 4

<channel> = <CHARACTER PROGRAM DATA>

CH1, ..., CH16 CH1 to CH16

ALL Total All Error

INSertion Total INS Error

OMISsion Total OMI Error

IOMission Total INS&OMI Error

<y1>,<m1>,<d1>,<h1>,<n1>,<s1> = <DECIMAL NUMERIC PROGRAM DATA>

Print start time (year/month/date/hour/minute/second)

<y2>,<m2>,<d2>,<h2>,<n2>,<s2> = <DECIMAL NUMERIC PROGRAM DATA>

Print end time (year/month/date/hour/minute/second)

Function Prints the recall text data for a specified unit on a specified CH in effect at a specified point of time (:...:TIME command).

Prints all the recall text data for a specified unit on a specified CH (:...:ALL command).

:SYSTem:PRINT:PATtern:TIME:ADDRes <unit>,<start>,<end>

:SYSTem:PRINT:PATtern:TIME:ALL <unit>

Parameter	<unit> = <DECIMAL NUMERIC PROGRAM DATA> 3 Unit inserted in slot 3 4 Unit inserted in slot 4 <start> = <NON-DECIMAL NUMERIC PROGRAM DATA> #H0 to <end> 0 to <end> bits/Step:1 (Print starting address: Specified in hex) <end> = <NON-DECIMAL NUMERIC PROGRAM DATA> <start> to Max <start> to Max bits/Step:1 (Print ending address: Specified in hex) Max: Program pattern ending address
Function	Prints addressed pattern data (type Time) for a specified unit on a specified CH (...:ADDRes command). Prints all pattern data (type Time) for a specified unit on a specified CH (...:ALL command).

:SYSTem:PRINT:PATtern:STATe:ADDRes <unit>,<start>,<end>

:SYSTem:PRINT:PATtern:STATe:ALL <unit>

Parameter	<unit> = <DECIMAL NUMERIC PROGRAM DATA> 3 Unit inserted in slot 3 4 Unit inserted in slot 4 <start> = <NON-DECIMAL NUMERIC PROGRAM DATA> #H0 to <end> 0 to <end> bits/Step:1 (Print starting address: Specified in hex) <end> = <NON-DECIMAL NUMERIC PROGRAM DATA> <start> to Max <start> to Max bits/Step:1 (Print ending address: Specified in hex) Max: Program pattern ending address
Function	Prints addressed pattern data (type State) for a specified unit on a specified CH (...:ADDRes command). Prints all pattern data (type State) for a specified unit on a specified CH (...:ALL command).

:SYSTem:PRINT:PATtern:DUMP:ADDReSS <unit>,<ch>,<form>,<start>,<end>

:SYSTem:PRINT:PATtern:DUMP:ALL <unit>,<ch>,<form>

Parameter	<unit> = <DECIMAL NUMERIC PROGRAM DATA> 3 Unit inserted in slot 3 4 Unit inserted in slot 4 <ch> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 CH1 to CH16/Step:1 0 The specified <unit> is a 1CH unit. <form> = <CHARACTER PROGRAM DATA> HEX Prints in hex. BIN Prints in binary. <start> = <NON-DECIMAL NUMERIC PROGRAM DATA> #H0 to <end> 0 to <end> bits/Step:1 (Print starting address: Specified in hex) <end> = <NON-DECIMAL NUMERIC PROGRAM DATA> <start> to Max <start> to Max bits/Step:1 (Print ending address: Specified in hex) Max: Program pattern ending address
Function	Prints addressed pattern data (type Dump) for a specified unit on a specified CH (...:ADDReSS command). Prints all pattern data (type Dump) for a specified unit on a specified CH (...:ALL command).

:SYSTem:PRINT:EMARgin

Function Prints Eye Margin measurement data.

:SYSTem:PRINT:REMargin

Function Prints Recall Eye Margin(Diagram) data

:SYSTem:PRINT:DELaY1

Function Prints Delay(Mode1) measurement data

:SYSTem:PRINT:DELaY2

Function Prints Delay(Mode2) measurement data.

6.6.3 STATus Subsystem

The STATus subsystem sets and queries SCPI-prescribed status registers and device-specific status registers.

:STATus:PRESet

Function Initializes status registers. The following registers are influenced by the execution of this command; see Chapter 5 for more details:

- SCPI-prescribed status register Transition Filter, Enable Register
- Device-specific status register Transition Filter, Enable Register

■ Setting and querying operation status registers

<Event> :STATus:OPERation[EVENT]?

<Condition> :STATus:OPERation:CONDition?

<Enable> :STATus:OPERation:ENABle <numeric>

:STATus:OPERation:ENABle?

<PTRansition> :STATus:OPERation:PTRansition <numeric>

:STATus:OPERation:PTRansition?

<NTRansition> :STATus:OPERation:NTRansition <numeric>

:STATus:OPERation:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits false
16	(BIT 4)	Error/alarm measurement in progress
256	(BIT 8)	An auto search in progress
512	(BIT 9)	Eye margin measurement in progress
1024	(BIT10)	Delay measurement in progress
8192	(BIT13)	INSTRument status register summary
Set the sum total of the bit position values of the bits to be set.		

Response <numeric> = <NR1 NUMERIC RESPNSE DATA>

Function Sets or queries the individual OPERation status register settings.

■ Setting and querying instrument status registers

<Event>	:STATus:OPERation:INSTrument[:EVENT]?
<Condition>	:STATus:OPERation:INSTrument:CONDition?
<Enable>	:STATus:OPERation:INSTrument:ENABle <numeric> :STATus:OPERation:INSTrument:ENABle?
<PTRansition>	:STATus:OPERation:INSTrument:PTRansition <numeric> :STATus:OPERation:INSTrument:PTRansition?
<NTRansition>	:STATus:OPERation:INSTrument:NTRansition <numeric> :STATus:OPERation:INSTrument:NTRansition?
Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 All bits false 4 (BIT 2) Measurement end 16 (BIT 4) Alarm change
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Sets or queries the individual INSTrument status register settings.

■ Setting and querying questionable status registers

<Event>	:STATus:QUEStionable[:EVENT]?
<Condition>	:STATus:QUEStionable:CONDition?
<Enable>	:STATus:QUEStionable:ENABle <numeric> :STATus:QUEStionable:ENABle?
<PTRansition>	:STATus:QUEStionable:PTRansition <numeric> :STATus:QUEStionable:PTRansition?
<NTRansition>	:STATus:QUEStionable:NTRansition <numeric> :STATus:QUEStionable:NTRansition?
Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 All bits false 512 (BIT 9) MONitor status register summary
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Sets or queries the individual QUEStionable status register settings.

■ Setting and querying MONitor status registers

<Event>	:STATus:QUESTionable:MONitor[:EVENT]?		
<Condition>	:STATus:QUESTionable:MONitor:CONDition?		
<Enable>	:STATus:QUESTionable:MONitor:ENABLE <numeric>		
	:STATus:QUESTionable:MONitor:ENABLE?		
<PTRansition>	:STATus:QUESTionable:MONitor:PTRansition <numeric>		
	:STATus:QUESTionable:MONitor:PTRansition?		
<NTRansition>	:STATus:QUESTionable:MONitor:NTRansition <numeric>		
	:STATus:QUESTionable:MONitor:NTRansition?		
Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>		
	0		All bits false
	1	(BIT 0)	Power failure
	2	(BIT 1)	SLOT1 status register summary
	8	(BIT 3)	SLOT3 status register summary
	16	(BIT 4)	SLOT4 status register summary
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>		
Function	Sets or queries the individual MONitor status register settings.		

■ Setting and querying SLOT1 status registers

<Event>	:STATus:QUESTionable:MONitor:SLOT1[:EVENT]?		
<Condition>	:STATus:QUESTionable:MONitor:SLOT1:CONDition?		
<Enable>	:STATus:QUESTionable:MONitor:SLOT1:ENABLE <numeric>		
	:STATus:QUESTionable:MONitor:SLOT1:ENABLE?		
<PTRansition>	:STATus:QUESTionable:MONitor:SLOT1:PTRansition <numeric>		
	:STATus:QUESTionable:MONitor:SLOT1:PTRansition?		
<NTRansition>	:STATus:QUESTionable:MONitor:SLOT1:NTRansition <numeric>		
	:STATus:QUESTionable:MONitor:SLOT1:NTRansition?		
Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>		
	0		All bits false
	1	(BIT 0)	PLL Unlock
	2	(BIT 1)	External Clock Loss
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>		
Function	Sets or queries the individual SLOT1 status register settings.		

■ Setting and querying SLOT3 status registers

<Event>	:STATus:QUESTionable:MONitor:SLOT3[:EVENT]?
<Condition>	:STATus:QUESTionable:MONitor:SLOT3:CONDition?
<Enable>	:STATus:QUESTionable:MONitor:SLOT3:ENABle <numeric> :STATus:QUESTionable:MONitor:SLOT3:ENABle?
<PTRansition>	:STATus:QUESTionable:MONitor:SLOT3:PTRansition <numeric> :STATus:QUESTionable:MONitor:SLOT3:PTRansition?
<NTRansition>	:STATus:QUESTionable:MONitor:SLOT3:NTRansition <numeric> :STATus:QUESTionable:MONitor:SLOT3:NTRansition?
Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 All bits false 32 (BIT 5) 200M 16CH PPG Clock Loss
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Sets or queries the individual SLOT3 status register settings.

■ Setting and querying SLOT4 status registers

<Event>	:STATus:QUESTionable:MONitor:SLOT4[:EVENT]?
<Condition>	:STATus:QUESTionable:MONitor:SLOT4:CONDition?
<Enable>	:STATus:QUESTionable:MONitor:SLOT4:ENABle <numeric> :STATus:QUESTionable:MONitor:SLOT4:ENABle?
<PTRansition>	:STATus:QUESTionable:MONitor:SLOT4:PTRansition <numeric> :STATus:QUESTionable:MONitor:SLOT4:PTRansition?
<NTRansition>	:STATus:QUESTionable:MONitor:SLOT4:NTRansition <numeric> :STATus:QUESTionable:MONitor:SLOT4:NTRansition?
Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 All bits false 2 (BIT 1) 200M 16CH ED status register summary 256 (BIT 8) ED Clock Loss 512 (BIT 9) FIFO Full 1024 (BIT 10) FIFO Empty 2048 (BIT 11) Phase Error
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Sets or queries the individual SLOT4 status register settings.

■ Setting and querying 200M 16CH (ED (slot 4) status registers

<Event> :STATus:QUESTionable:MONitor:SLOT4:M2CH16[:EVENT]?
 <Condition> :STATus:QUESTionable:MONitor:SLOT4:M2CH16:CONDition?
 <Enable> :STATus:QUESTionable:MONitor:SLOT4:M2CH16:ENABle <numeric>
 :STATus:QUESTionable:MONitor:SLOT4:M2CH16:ENABle?
 <PTRansition> :STATus:QUESTionable:MONitor:SLOT4:M2CH16:PTRansition <numeric>
 :STATus:QUESTionable:MONitor:SLOT4:M2CH16:PTRansition?
 <NTRansition> :STATus:QUESTionable:MONitor:SLOT4:M2CH16:NTRansition <numeric>
 :STATus:QUESTionable:MONitor:SLOT4:M2CH16:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
 0 All bits false
 1 (BIT 0) CH1&CH2 status register summary
 2 (BIT 1) CH3&CH4 status register summary
 4 (BIT 2) CH5&CH6 status register summary
 8 (BIT 3) CH7&CH8 status register summary
 16 (BIT 4) CH9&CH10 status register summary
 32 (BIT 5) CH11&CH12 status register summary
 64 (BIT 6) CH13&CH14 status register summary
 128 (BIT 7) CH15&CH16 status register summary

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries the individual 200M 16CH ED (slot 4) status register settings.

■ Setting and querying 200M CH1&CH2 (slot 4) status registers

<Event>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2[:ENABLE]?

<Condition>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2:CONDition?

<Enable>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2:ENABLE <numeric>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2:ENABLE?

<PTRansition>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2:PTRansition <numeric>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2:PTRansition?

<NTRansition>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2:NTRansition <numeric>

:STATus:QUESTionable:MONitor:SLOT4:M2CH16:CH1_2:NTRansition?

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits false
1	(BIT 0)	CHX INS error
2	(BIT 1)	CHX OMI error
4	(BIT 2)	CHX ALL error
8	(BIT 3)	CHX Pattern Sync Loss(PRBS)
16	(BIT 4)	CHX Pattern Sync Loss(PRGM)
32	(BIT 5)	CHX Clock Loss
64	(BIT 6)	CHX+1 INS error
128	(BIT 7)	CHX+1 OMI error
256	(BIT 8)	CHX+1 ALL error
512	(BIT 9)	CHX+1 Pattern Sync Loss(PRBS)
1024	(BIT10)	CHX+1 Pattern Sync Loss(PRGM)
2048	(BIT11)	CHX+1 Clock Loss

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function

Sets or queries the individual 200M CH1&CH2 (slot4) status registers settings.

■ Setting or querying 200M CH3&CH4 (slot 4) to 200M CH15&CH16 (slot 4) status registers

These commands are similar to those for the 200M CH1&CH2 (slot 4) status registers described above, except that the headers are changed as shown below. The parameters and responses are identical.

. CH1_2 → CH3_4, CH5_6, ..., CH15_16

Chapter 7 Sample Programs

7.1 Programming Environment

This section introduces sample programs that work with the MP1630B. These programs were created by assuming that an IBM-PC/AT or compatible is used as a host controller and that it controls this instrument via its RS-232C interface. It was also assumed that a 200M CG, a 200M 16 CH PPG, and a 200M 16CH ED unit were inserted in slots 1, 3, and 4, respectively.

The sample programs were written in Microsoft® Visual Basic® Programming System Professional Edition version 3.0 and were validated on DECpc, Microsoft® Windows® Operating System version 3.1, and Visual Basic® Version 3.0.

■ Preparations for the running sample programs

Assume that the controller's floppy disk drive is A, its hard disk drive is C, and the sample programs are installed in the directory C:\MP1630RS.

1. Set up the instrument and the controller's RS-232C interface.
<Example>
 - Speed : 9600, Parity : None, Data Bit Length : 8bits,
Stop Bit Length : 1bits, Flow Control : Hardware
2. Install a sample program on the controller's hard disk drive. Choose [Icon]-[Run] from the program manager, type "A:\setup" into the command line text box, and click the OK button. The sample program will be copied to the hard disk drive.
3. Unless otherwise specified, the sample programs are copied into the directory C:\MP1630RS. The "MP1630B Remote Sample Program" group and a sample program icon are created.
4. To run the sample program, click the Sample Program icon in the MP1630B Remote Sample Program group box from the program manager. When reading the program from Visual Basic®, read C:\MP1630RS\SAMPLE.MAK.

Note

These sample programs have been created on the basis of the assumption that they are run on Visual Basic®. The programs cannot be read and referenced as a text file when the controller does not support Visual Basic®.

7.2 Program Descriptions

Simple ways to run the sample programs are described below. Choose one of the following five sample programs from the menu on display:

1. 200M CG Setup
2. 200M 16CH PPG Setup
3. 200M 16CH ED Setup
4. Error/Alarm Measurement
5. Status Register Setup

■ Initialize

When Click the Initialize button that located parent window, sending following commands for initialize.

1. :SYSTem:MEMory:INITialize : Reset to the factory defaults
2. :System:BEEper:SYSTem:SET OFF : Set to Buzzer OFF
:System:BEEper>Error:SET OFF
:System:BEEper:ALARm:SET OFF
3. *CLS : Clear the status register events and the error/event queue.

Before execute these five sample programs, you need initialize by this Initialize button.

■ Watch window

All output data as commands and all input data as response are displayed Watch window.

■ Common function

These five programs have following common function.

1. Following the command transmission above, transmit the :SYSTem:ERRor? command and display the result of its execution in the Error Status text box. (The sample programs do not check constraints on settings. If a value out of range is entered, an error will be indicated in the Error Status text box after the transmission of the command.)

■ 200M CG Setup

This sample program sets up the 200M CG unit.

<Instructions>

1. Choose [Select]-[200M CG Setup...] from the sample program menu. The 200M CG Setup screen will open.
2. Set a reference frequency and a frequency for internal clock generation.
3. Press the Set button to transmit the settings entered into the screen to the CG unit installed in the instrument. A PLL Unlock check is made every second. If an alarm is present, the PLL Unlock lamp lights. The alarm query is iterated until the Exit button is pressed.
4. Press the Exit button to close the screen.

Note

With this program, any settings that are not entered into the screen are set to their factory defaults.

■ 200M 16CH PPG Setup

This sample program sets up the 200M 16CH PPG unit.

<Instructions>

1. Choose [Select]-[200M 16CH PPG Setup...] from the sample program menu. The 200M 16CH PPG Setup screen will open.
2. Set the polarity of the clock output interface, plus the amplitude if the Level setting is VAR.
3. Set the skew of the data output interface, plus the amplitude if the Level setting is VAR.
4. Set the number of stages of the PRBS pattern, and the mark ratio.
5. Press the Set button to transmit the settings entered into the screen to the PPG unit.
6. Press the Exit button to close the screen.

■ 200M 16CH ED Setup

This sample program sets up the 200M 16CH ED unit.

1. Choose [Select]-[200M 16CH ED Setup...] from the sample program menu. The 200M 16CH ED Setup screen will open.
2. Set the polarity of the clock output interface.
3. Set the threshold of the data output interface if the Level setting is VAR.
4. Set the number of stages of the PRBS pattern, and the mark ratio.
5. Press the Set button to transmit the settings entered into the screen to the ED unit.
6. Press the Exit button to close the screen.

■ Error/Alarm Measurement

This sample program executes error/alarm measurement and displays the measurement results.

<Instructions>

1. Choose [Select]-[Error/Alarm Measurement...] from the sample program menu. The Error/Alarm Measurement screen will open.
2. Set the measurement mode and cycle.
3. Press the Set button. The Measurement lamp lights and measurement starts. The mode of error detection in effect at this time is CH ALL. When the measurement is ended, the ER over each CH is displayed in the Result text box. Press the Stop button to force the measurement to an end.
4. Press the Exit button to close the screen.

Note

With this program, any settings that are not entered into the screen are set to their factory defaults.

■ Status Register Setup

This sample program executes sets up status registers and monitors events (errors or alarms).

<Instructions>

1. Choose [Select]-[Status Register Setup...] from the sample program menu. The Status Register Setup screen will open.
2. Select the event to monitor. (Only CH1 events can be monitored.)
3. Press the Set button. Error/alarm measurement starts cyclically at 1 second intervals, monitoring events. When an event occurs, the Event lamp lights; it goes off when the event no longer exists. Press the Stop button to terminate the measurement and monitoring.
4. Press the Exit button to close the screen.

Note

With this program, any settings that are not entered into the screen are set to their factory defaults.

Appendix

Appendix A SCPI Conformance Information

Appendix B SCPI Error Messages

Appendix C Command List

Appendix A SCPI Conformance Information

■ SCPI version

The MP1630B conforms to the SCPI version 1995.0.

■ SCPI-defined commands

Among all the commands supported by instrument, the following are defined by SCPI 1995.0:

:SYSTem	*IDN?
:DATE	*RST
:DATE?	*TST?
:TIME	*OPC
:TIME?	*OPC?
:ERRor?	*WAI
:VERSion?	*CLS
	*ESE
:STATus	*ESE?
:PRESet	*ESR?
:OPERation	*SRE
[:EVENT]?	*SRE?
:CONDtion?	*STB?
:ENABle	*TRG
:ENABle?	*OPT?
:PTRansition	
:PTRansition?	
:NTRansition	
:NTRansition?	
:INSTrument	
[:EVENT]?	
:CONDtion?	
:ENABle	
:ENABle?	
:PTRansition	
:PTRansition?	
:NTRansition	
:NTRansition?	
:QUEStionable	
[:EVENT]?	
:CONDtion?	
:ENABle	
:ENABle?	
:PTRansition	
:PTRansition?	
:NTRansition	
:NTRansition?	

■ Non-SCPI commands

Among all the commands supported by instrument, the following are not defined by SCPI:

:SOURce	:SOURce
:PATtern	:PATtern
:MODE	[:NORMal]
:MODE?	:BURSt
[:NORMal]	:MODE
:OMODE	:MODE?
:OMODE?	:CYCLe
:TYPE	:CYCLe?
:TYPE?	:ELENgth
:PRBS	:ELENgth?
:MRATio	:PMODE
:MRATio?	:PMODE?
:BSHift	:PDIFference
:BSHift?	:PHASe
:ZSUBstitute	:PHASe?
:LENgth	:PON
:LENgth?	:ECHannel
:ZLENgth	:ECHannel?
:ZLENgth?	:PLENgth
:LOGic	:PLENgth?
:LOGic?	:PROGram
:PROGram	:LOGic
:LENgth	:LOGic?
:LENgth?	:PRBS
:LOGic	:TYPE
:LOGic?	:TYPE?
:MIX	:MRATio
:PROGram	:MRATio?
:LOGic	:BSHift
:LOGic?	:BSHift?
:PRBS	:BLOCK
:TYPE	:SIZE
:TYPE?	:SIZE?
:MRATio	:SET
:MRATio?	:SET?
:BSHift	:LENgth
:BSHift?	:LENgth?
:BLOCK	
:SIZE	
:SIZE?	
:SET	
:SET?	
:LENgth	
:LENgth?	

:SOURce

:PATtern:

:PON

:OTIMing

:BMODE

:BMODE?

:BCYCLE

:BCYCLE?

:GTIME

:GTIME?

:DATA

:CHANnel

:CHANnel?

:BLOCK

:BLOCK?

:BDATa

:CHANnel

:CHANnel?

:BLOCK

:BLOCK?

:DREVerse

:CHANnel

:BLOCK

:ADDRESS

:DELTA

:ADDRESS

:CHANnel?

:BLOCK?

:DELAy

:CYCLE

:CYCLE?

:TYPE

:TYPE?

:SOURce

:PATtern

:EADDITION

:SET

:SET?

:SINGLE

:MODE1

:MODE1?

:MODE2

:MODE2?

:MODE3

:MODE3?

:RATE

:RATE?

:BCYCLE

:BCYCLE?

:ELENgth

:ELENgth?

:ADDITION

:ADDITION?

:ABLOCK

:ABLOCK?

:BLOCK

:BLOCK?

:OUTPut

:SET

:SET?

:CLOCK

:POLarity

:POLarity?

:OMODE

:OMODE?

:OUTPut

:OUTPut?

:DELAy

:DELAy?

:LEVel

:LEVel?

:TERMination

:TERMination?

:AMPLitude

:AMPLitude?

:OFFSet

:OFFSet?

:OREFERENCE

:OREFERENCE?

:OUTPut

:DATA

:MODE
 :MODE?
 :OUTPut
 :OUTPut?
 :SKEW
 :SKEW?
 :LEVel
 :LEVel?
 :TERMination
 :TERMination?
 :AMPLitude
 :AMPLitude?
 :OFFSet
 :OFFSet?
 :OREFERENCE
 :OREFERENCE?

:AUXiliary

:LEVel
 :LEVel?
 :CHANnel
 :CHANnel?
 :TYPE
 :TYPE?
 :LOGic
 :LOCic?
 :BLOCation
 :BLOCation?
 :ELOCation
 :ELOCation?
 :ENVelop
 :PDELay
 :PDELay?
 :NDELay
 :NDELay?
 :RESet
 :PWIDth
 :PWIDth?
 :LOCation
 :LOCation?

:SYNC

:SOURce
 :SOURce?
 :PATtern
 :CHANnel
 :CHANnel?
 :PATtern
 :PATtern?

:OUTPut

:SYNC

:CLOCK
 :CHANnel
 :CHANnel?

:TRIGger

:SOURce
 :SOURce?
 :CHANnel
 :CHANnel?
 :BLOCK
 :BLOCK?

:BTRigger

:DELay
 :DELay?

:INSTrument

:COUPle

:PATtern
 :SET
 :SET?

:SENSe

:PATTern

:MODE

:MODE?

[:NORMal]

:IMODE

:IMODE?

:TYPE

:TYPE?

:PRBS

:MRATio

:MRATio?

:BSHift

:BSHift?

:ZSUBstitute

:LENGth

:LENGth?

:ZLENGth

:ZLENGth?

:LOGic

:LOGic?

:PROGram

:LENGth

:LENGth?

:LOGic

:LOGic?

:MIX

:PROGram

:LOGic

:LOGic?

:PRBS

:TYPE

:TYPE?

:MRATio

:MRATio?

:BSHift

:BSHift?

:BLOCK

:SIZE

:SIZE?

:SET

:SET?

:LENGth

:LENGth?

::SENSe

:PATTern

[:NORMal]

:SYNC

:ASYNc

:PROGram

:PROGram?

:PRBS

:PRBS?

:THReshold

:PROGram

:PROGram?

:PRBS

:PRBS?

:PSMode

:PSMode?

:BSMode

:BSMode?

:FLENGth

:FLENGth?

:BURSt

:ELENGth

:ELENGth?

:PMODE

:PMODE?

:SENSe

:PATtern

:PON

:EChanneL

:EChanneL?

:PROGram

:LOGic

:LOGic?

:PRBS

:TYPE

:TYPE?

:MRATio

:MRATio?

:BSHift

:BSHift?

:BLOCk

:SIZE

:SIZE?

:SET

:SET?

:LENGth

:LENGth?

:SYNC

:ASYNc

:PROGram

:PROGram?

:PRBS

:PRBS?

:THReshold

:PROGram

:PROGram?

:PRBS

:PRBS?

:BSMode

:BSMode?

:FLENGth

:FLENGth?

:DATA

:CHANnel

:CHANnel?

:BLOCk

:BLOCk?

:SENSe

:PATtern

:BDATa

:CHANnel

:CHANnel?

:BLOCk

:BLOCk?

:DREVerse

:CHANnel

:BLOCk

:ADDReSS

:DELTA

:ADDReSS

:CHANnel?

:BLOCk?

:MEASure

:TEST

:TEST?

:START

:STOP

:EALarm

:MODE

:MODE?

:PERiod

:PERiod?

:BTIME

:SET

:SET?

:START

:START?

:ERRor

:MODE

:MODE?

:DBLock

:DBLock?

:BLOCk

:BLOCk?

:START?

:STOP?

:STATe?

:ELAPsed?

:TIMed?

:ITIME?

:SENSe

:MEASure

:DELaY1

:MODE
:MODE?
:TOUT
:TOUT?
:CHANnel
:CHANnel?
:STATe?
:MSTate?
:STARt?
:STOP?
:ELAPsed?
:TYPE
:TYPE?

:DELaY2

:MODE
:MODE?
:TOUT
:TOUT?
:PATtern
:PATtern?
:CHANnel
:CHANnel?
:STATe?
:MSTate?
:STARt?
:STOP?
:ELAPsed?
:TYPE
:TYPE?

:EMARgin

:MODE
:MODE?
:TYPE
:TYPE?
:MARGin
:THReshold
:THReshold?
:RESolution
:RESolution?
:DIAGram
:THReshold
:THReshold?
:POINT?

:SENEe

:MEASure

:EMARgin

:UNIT
:UNIT?
:CHANnel
:CHANnel?
:STATe?
:STARt?
:STOP?
:ELAPsed?

:ASEarch

:MODE
:MODE?
:CHANnel
:CHANnel?
:STARt
:STOP
:PATtern
:PATtern?
:STATe?
:FCHannel?

:RDIagram

:CHANnel?
:THReshold?
:POINT?
:STARt?
:STOP?

:ANALyze

:CHANnel
:CHANnel?
:TTEXT
:DATA
:DATA?
:IDATa
:ITEM
:ITEM?
:PERiod
:PERiod?
:EDATa
:ITEM
:ITEM?
:SDATa
:THReshold
:THReshold?

:INPut

:CLOCK

:MODE
:MODE?
:IMODE
:IMODE?
:POLarity
:POLarity?
:DELay
:DELay?
:TERMination
:TERMination?

:DATA

:MODE
:MODE?
:LEVel
:LEVel?
:THReshold
:THReshold?
:TERMination
:TERMination?

:RCLock

:SOURce
:SOURce?

:CALCulate

:EALarm

:SLEValuation
:SLEValuation?
:CLEValuation
:CLEValuation?
:ERRor
:INTerval
:INTerval?
:PERFormance
:MEASurement
:MEASurement?
:THReshold
:THReshold?
:TEINTerval
:MEASurement
:MEASurement?
:PFAil
:PFAil?

:CALCulate

:DATA

:EALarm?
:EMARgin?
:DELay1?
:DELay2?
:STARt?
:SIZE?
:TEXT?
:MONitor?
:STORe
:CLEar

:OUTPut

:CLOCK

:SOURce
:SOURce?
:FREQuency
:FREQuency?

:RCLock

:SElect
:SElect?

:JITTer

:EMODulation
:EMODulation?
:RANGe
:RANGe?

:ARM

:SOURce
:SOURce?
:SET
:SET?
:STARt

:DISPlay

```

:ACHannel
:ACHannel?
:WINDow
    :OPEN
    :OPEN?
    :CLOSe
:SYSTem
    [:NAME]
    [:NAME]?
:SETup
    [:NAME]
    [:NAME]?
:CLOCK
    :CHANnel
    :CHANnel?
:DATA
    :CHANnel
    :CHANnel?
:PATtern
    :PAGE
    :PAGE?
    :CHANnel
    :CHANnel?
:AUXiliary
    :PORT
    :PORT?
:TEST
    [:NAME]
    [:NAME]?
:EALarm
    :CHANnel
    :CHANnel?
:EADdition
    :CHANnel
    :CHANnel

```

:DISPlay

```

:RESult
:TIME
:TIME?
:EALarm
    [:NAME]
    [:NAME]?
:MODE
:MODE?
:TYPE
:TYPE?
:PITem
:PITem?
:TITem
:TITem?
:UNIT
:UNIT?
:CHANnel
:CHANnel?
:ZOOM1|:ZOOM2|
:ZOOM3|:ZOOM4
    :SET
    :SET?
    :UNIT
    :UNIT?
    :CHANnel
    :CHANnel?
:ITEM
:ITEM?
:PITem
:PITem?
:TITem
:TITem?
:HRESet
:EMARgin
:MODE
:MODE?
:ERATe
:ERATe?
:SCALE
    :AScale
    :VOLTage
    :VOLTage?
    :PHASe
    :PHASe?

```

```

:DISPlay
  :RESult
    :EMARgin
      :MARKer
        :SET
        :SET?
        :MODE
        :MODE?
        :ERATe
        :ERATe?
        :MOVE
        :POSition?
    :CUSTomize
      :SETup1|:SETup2|
      :SETup3|:SETup4|
      :SETup5|:SETup6
        :UNIT
        :UNIT?
        :CHANnel
        :CHANnel?
        :ITEM
        :ITEM?
      :PATtern
        :UNIT
        :UNIT?
        :CHANnel
        :CHANnel?
        :OFFSet
        :OFFSet?
        :BLOCK
        :BLOCK?
      :RESult
        :UNIT
        :UNIT?
        :CHANnel
        :CHANnel?
        :ITEM
        :ITEM?
    :ANALyze
      :MODE
      :MODE?
      :UNIT
      :UNIT?
      :CHANnel
      :CHANnel?

```

```

:DISPlay
  :ANALyze
    :TEXT
      :TIME
      :TIME?
      :SCRoll
    :SYSTem
      :BEEPer
      :ERRor
        :SET
        :SET?
        :CHANnel
        :CHANnel?
      :ALARm
        :SET
        :SET?
        :CHANnel
        :CHANnel?
      :SYSTem
        :SET
        :SET?
        :TYPE
        :TYPE?

```

:SYSTem

:MODE?

:BSIZe?

:ORGanization

:HARDware?

:SOFTware?

:MSVersion?

:MMEMory

:RECall

:STORe

:QRECall

:QSTore

:CATalog?

:MEMory

:INITialize

:PRINt

:COPY

:TEXT

:EALarm

:TTEXT

:TIME

:ALL

:RTEXT

:TIME

:ALL

:PATtern

:TIME

:ADDRESS

:ALL

:STATe

:ADDRESS

:ALL

:DUMP

:ADDRESS

:ALL

:EMARgin

:REMargin

:DELay1

:DELay2

:STATus

:QUEStionable

:MONitor

[:EVENT]?

:CONDtion?

:ENABLe

:ENABLe?

:PTRansition

:PTRansition?

:NTRansition

:NTRansition?

:SLOT1

[:EVENT]?

:CONDtion?

:ENABLe

:ENABLe?

:PTRansition

:PTRansition?

:NTRansition

:NTRansition?

:STATus

:QUESTionable

:MONitor

:SLOT3|:SLOT4

[:EVENT]?

:CONDtion?

:ENABle

:ENABle?

:PTRansition

:PTRansition?

:NTRansition

:NTRansition?

:M2CH16

[:EVENT]?

:CONDtion?

:ENABle

:ENABle?

:PTRansition

:PTRansition?

:NTRansition

:NTRansition?

:CH1_2|:CH3_4|:CH5_6|:CH7_8|

:CH9_10|:CH11_12|:CH13_14|:CH15_16

[:EVENT]?

:CONDtion?

:ENABle

:ENABle?

:PTRansition

:PTRansition?

:NTRansition

:NTRansition?

Appendix B SCPI Error Messages

SCPI defines error-specific codes and messages as responses to the SCPI command :SYSTem:ERRor?.

A detailed definition of the error messages supported by this instrument follows.

■ Command errors

The error codes -199 and -100 indicate that an IEEE488.2 syntax error occurred. Bit 5 of the device's event status register is set at the same time.

An error occurs when any one of the following events arises:

- The device received a message in violation of the specifications of IEEE488.2.
- A device-specific command or a header in violation of the specifications of common commands was received.
- GET (Group Execute Trigger) was transmitted within a program message.

Table B-1 Error Codes and Error Messages in Command Errors (1/2)

Code	Message	Error detection condition
-101	Invalid character	An invalid character is found in a header or parameter. Examples: # is included in a header. :SOURce3:PATtern:M#DE NORMal \$ is included in a parameter. :SENSe3:PATtern:MODE N\$RMAL
-104	Data type error	The parameter type does not match its specification. Example: Character data was transmitted to a command having a parameter of the type numeric.
-105	GET not allowed	GET was transmitted within a program message.
-108	Parameter not allowed	Too many parameters were specified. Example: Four parameters were transmitted when only two were required.
-112	Program mnemonic too long	A program mnemonic of 12 characters or longer was entered. Example: A header colon is missing. :SOURce3:PATtern:ZSUBstituteLENGth 1,7
-113	Undefined header	The header was valid in its syntax but was not defined by the device. Example: A header was written incorrectly. :SOURce3:PATtern:MDE NORMal
-120	Numeric data error	There is an error in numeric data.
-121	Invalid character in number	An invalid character is found in numeric data. Example: A symbol is included in the numeric data. :SOURce3:PATtern:TYPE #,PRBS7
-130	Suffix error	There is an error in a suffix. Example: Part of the parameter of the type numeric was romanized by mistake. :OUTPut3:CLOCK:FREQuency 100I

Table B-1 Error Codes and Error Messages in Command Errors (2/2)

Code	Message	Error detection condition
-144	Character data too long	Character data of 12 characters or longer was entered.
-150	String data error	Undefined string data. Example: A single quotation mark and a double quotation mark were intermixed. :CALCulate3:DATA:EALarm "ER:INS'

■ Execution errors

The error codes -299 and -200 indicate that an error occurred in the executive control of the device. Bit 4 of the device's event status register is set at the same time.

An error occurs when any one of the following events arises:

- The <PROGRAM DATA> that follows a header is undefined by the device.
- A program message is not executable due to particular device status.

Table B-2 Error Codes and Error Messages in Execution Errors

Code	Message	Error detection condition
-220	Parameter error	There is an error in the parameter.
-221	Setting conflict	The parameter is valid but cannot be set due to particular device status. Example: An attempt was made to set a PON pattern when the pattern mode was Normal.
-222	Data out of range	Numeric data is out of range for the device
-223	Too much data	String data exceeds the defined length.
-224	Illegal parameter value	The received parameter is illegal. Examples: Character data is illegal. String data is illegal.
-232	Invalid format	Specified file is invalid format. This error occurs when you open PPG file as Pattern file.
-233	Invalid version	Specified file is saved by application software whose version is newer.
-241	Hardware missing	The command cannot be executed because the option is not installed.
-252	Missing media	The floppy disk is not inserted.
-253	Corrupt media	The floppy disk is not formatted.
-254	Media full	The hard disk or floppy disk does not have enough space available.
-255	Directory full	The hard disk or floppy disk directory does not have enough space available.
-256	File name not found	The file is not found on the hard disk or floppy disk.
-257	File name error	The filename is invalid.
-258	Media protected	Data cannot be written to the floppy disk because it is write-protected.

■ Device-specific errors

The error codes -399 and -300 indicate that the device has detected an error other than command, query, and execution errors. Such errors might convey hardware or firmware failures or self-test errors.

When device-specific errors occur, bit 3 of the device's event status register is set.

Table B-3 Error Codes and Error Messages in Device-Specific Errors

Code	Message	Error detection condition
-310	System error	A system error occurred.
-315	Configuration memory lost	Resume memory was lost.

■ Query errors

The error codes -499 and -400 indicate that a message switching control protocol error occurred under the device's output queue control. When query errors occur, bit 2 of the device's event status register is set.

An error occurs when any one of the following events arises:

- An attempt was made to read from the output queue when there was no output.
- Data on the output queue was lost.

Table B-4 Error Codes and Error Messages in Query Errors

Code	Message	Error detection condition
-410	Query INTERRUPTED	An interrupt due to a new command occurred before the device completed the transmission of a response message. Example: *TRG was transmitted before the complete transmission of a response.
-420	Query UNTERMINATED	No query had been transmitted corresponding to the response message to be read, or the query is not terminated completely.
-430	Query DEADLOCKED	Buffering of more data than was permitted by the storage area space available was attempted. Example: The device cannot proceed with measurement any farther because both the input and output buffers are full.

Appendix C Command List

This appendix lists the commands supported by this instrument, grouped by window. Query commands are not stated (they are stated, however, when a corresponding program command does not exist).

See Chapter 6 for more command details, including parameter types and contents.

■ Parent window

Table C-1 Parent Window Command List (1/2)

Setup item		Command
File menu	Quick Open	:SYSTem:MMEMory:QRECall
	Quick Save	:SYSTem:MMEMory:QSTore
	Open	:SYSTem:MMEMory:RECall
	Save	:SYSTem:MMEMory:STORE
	Hard Copy	:SYSTem:PRINt:COpy
	Print...	:SYSTem:PRINt:EALarm
		:SYSTem:PRINt:TTEXT:TIME
		:SYSTem:PRINt:TTEXT:ALL
		:SYSTem:PRINt:RTEXT:TIME
		:SYSTem:PRINt:RTEXT:ALL
		:SYSTem:PRINt:PATtern:TIME:ADDress
		:SYSTem:PRINt:PATtern:TIME:ALL
		:SYSTem:PRINt:PATtern:STATe:ADDress
		:SYSTem:PRINt:PATtern:STATe:ALL
		:SYSTem:PRINt:PATtern:DUMP:ADDress
		:SYSTem:PRINt:PATtern:DUMP:ALL
		:SYSTem:PRINt:EMARgin
		:SYSTem:PRINt:REMargin
		:SYSTem:PRINt:DELay1
		:SYSTem:PRINt:DELay2
	Initialize	:SYSTem:MEMory:INITialize
View menu	Display ALL CH	:DISPlay:ACHannel
Windowmenu	Tile	
	Cascade	
	Window name	
Tool ber		:OUTPut:SET

Table C-1 Parent Window Command List (2/2)

Setup item		Command
Subwindow manipulations	Open	:DISPlay:WINDow:OPEN
	Close	:DISPlay:WINDow:CLOSe
Others	Disk information query	:SYSTem:MMEMory:CATalog?
	Optional character string print	:SYSTem:PRINt:TEXT

■ System subwindow

Table C-2 System Subwindow Command List

Setup item	Command
Panel switching	:DISPlay:SYSTem[:NAME]

● System:System panel

Table C-3 System:System Panel Command List

Setup item		Command
System Mode		:SYSTem:MODE?
Buzzer (System Error)	ON/OFF status setting	:SYSTem:BEEPer:SYSTem:SET
	Item setting	:SYSTem:BEEPer:SYSTem:TYPE
Buzzer (Measruement)	Error	:SYSTem:BEEPer:ERRor:SET
	Alarm	:SYSTem:BEEPer:ALARm:SET
Date & Time	Date	:SYSTem:DATE
	Time	:SYSTem:TIME

● System:Option panel

Table C-4 System:Option Panel Command List

Setup item	Command
Hardware	:SYSTem:ORGAnization:HRADware?
Software	:SYSTem:ORGAnization:SOFTware?

■ Setup subwindow

Table C- 5 Setup Subwindow Command List

Setup item	Command
Panel switching	:DISPlay:SETup[:NAME]

● Setup:Setup panel

Table C- 6 Setup:Setup Panel Command List (200M 16CH PPG) (1/2)

Setup item	Command
Pattern Mode	:SOURce:PATtern:MODE
Clock&Data Connection	Data Output Mode :OUTPut:DATA:MODE
Common Setting	Pattern :INSTrument:COUPle:PATtern:SET

Table C- 6 Setup:Setup Panel Command List (200M 16CH ED) (2/2)

Setup item	Command
Pattern Mode	:SENSe:PATtern:MODE
Clock&Data Connection	Clock Input Mode :INPut:CLOCK:MODE Data Input Mode :INPut:DATA:MODE Reference Clock :INPut:RClock:SOURce
Common Setting	Pattern :INSTrument:COUPle:PATtern:SET

● Setup:Frequency panel

Table C- 7 Setup:Frequency Panel Command List

Setup item	Command
Clock	Clock Source :OUTPut:CLOCK:SOURce Reference :OUTPut:RClock:SElect Frequency :OUTPut:CLOCK:FREQuency
Arming	Source :ARM:SOURce Arming ON/OFF :ARM:SET Arming start :ARM:STARt
EXT Modulation	:OUTPut:JITTer:EMODulation
Jitter Monitor	MonitorRange :OUTPut:JITTer:RANGe
Alarm monitor	:CALCulate:DATA:MONitor?

- Setup:Clock I/F panel

Table C- 8 Setup:Clock I/F Panel Command List (1/3) (200M 16CH PPG)

Setup item		Command
Polarity		:OUTPut:CLOCK:POLarity
Output Mode		:OUTPut:CLOCK:OMODE
Output		:OUTPut:CLOCK:OUTPut
Delay		:OUTPut:CLOCK:DELay
Level		:OUTPut:CLOCK:LEVel
Termination of DUT		:OUTPut:CLOCK:TERMination
Amplitude		:OUTPut:CLOCK:AMPLitude
Offset	Offset value	:OUTPut:CLOCK:OFFSet
	Offset reference	:OUTPut:CLOCK:OREference

Table C- 8 Setup:Clock I/F Panel Command List (2/3) (200M 16CH ED)

Setup item		Command
Polarity		:INPut:CLOCK:POLarity
Input Mode		:INPut:CLOCK:IMODE
Delay		:INPut:CLOCK:DELay
Termination		:INPut:CLOCK:TERMination

Table C- 8 Setup:Clock I/F Panel Command List (3/3)

Setup item	Command
Channel switching	:DISPlay:SETup:CLOCK:CHANnel

- **Setup:Data I/F panel**

Table C- 9 Setup:Data I/F Panel Command List (1/3) (200M 16CH PPG)

Setup item		Command
Output		:OUTPut:DATA:OUTPut
Skew		:OUTPut:DATA:SKEW
Level		:OUTPut:DATA:LEVel
Termination of DUT		:OUTPut:DATA:TERMination
Amplitude		:OUTPut:DATA:AMPLitude
Offset	Offset value	:OUTPut:DATA:OFFSet
	Offset reference	:OUTPut:DATA:OREFERENCE

Table C- 9 Setup:Data I/F Panel Command List (2/3) (200M 16CH ED)

Setup item		Command
Level		:INPut:DATA:LEVel
Threshold		:INPut:DATA:THReshold
Termination		:INPut:DATA:TEMination

Table C- 9 Setup:Data I/F Panel Command List (3/3) (200M 16CH PPG)

Setup item		Command
Channel switching		:DISPlay:SETup:DATA:CHANnel

- Setup:Pattern panel

Table C-10 Setup:Pattern Panel Command List (1/7)
(200M 16CH PPG/Pattern Mode: Normal)

Setup item		Command
Output Pattern		:SOURce:PATtern[:NORMal]:OMODE
Pattern Setting	Pattern	:SOURce:PATtern[:NORMal]:TYPE
Pattern Setting (PRBS)	Mark Ratio	:SOURce:PATtern[:NORMal]:PRBS:MRATio
	Bit Shift	:SOURce:PATtern[:NORMal]:PRBS:BSHift
Pattern Setting (Zero-Subst)	Pattern Length	:SOURce:PATtern[:NORMal]:ZSUBstitute:LENGth
	Zero-Length	:SOURce:PATtern[:NORMal]:ZSUBstitute:ZLENGth
	Logic	:SOURce:PATtern[:NORMal]:ZSUBstitute:LOGic
Pattern Setting (PRGM)	Pattern Length	:SOURce:PATtern[:NORMal]:PROGram:LENGth
	Logic	:SOURce:PATtern[:NORMal]:PROGram:LOGic
Pattern Setting (MIX)	PRGM(Logic)	:SOURce:PATtern[:NORMal]:MIX:PROGram:LOGic
	PRBS(Pattern)	:SOURce:PATtern[:NORMal]:MIX:PRBS:TYPE
	PRBS(Mark Ratio)	:SOURce:PATtern[:NORMal]:MIX:PRBS:MRATio
	PRBS(Bit Shift)	:SOURce:PATtern[:NORMal]:MIX:PRBS:BSHift
	Block(Row, Column)	:SOURce:PATtern[:NORMal]:MIX:BLOCK:SIZE
	Block(Detail Setup)	:SOURce:PATtern[:NORMal]:MIX:BLOCK:SET
	Block (Column Length)	:SOURce:PATtern[:NORMal]:MIX:BLOCK:LENGth
Phase Difference		:SOURce:PATtern[:NORMal]:PDifference:PHASe
Burst Setting	Mode	:SOURce:PATtern[:NORMal]:BURSt:MODE
	Burst Cycle	:SOURce:PATtern[:NORMal]:BURSt:CYCLe
	Enable Length	:SOURce:PATtern[:NORMal]:BURSt:ELENGth
	PRGM Data Mode	:SOURce:PATtern[:NORMal]:BURSt:PMODE

Table C-10 Setup:Pattern Panel Command List (2/7)
(200M 16CH PPG/Pattern Mode: PC N)

Setup item		Command
Enable CHs		:SOURce:PATtern:PON:ECHannel
PR	Length	:SOURce:PATtern:PON:PLENght
OH/Information Pattern	PRGM(Logic)	:SOURce:PATtern:PON:PROGram:LOGic
	PRBS(Pattern)	:SOURce:PATtern:PON:PRBS:TYPE
	PRBS(Mark Ratio)	:SOURce:PATtern:PON:PRBS:MRATio
	PRBS(Bit Shift)	:SOURce:PATtern:PON:PRBS:BSHift
	Block(Number of blocks)	:SOURce:PATtern:PON:BLOCK:SIZE
	Block(Detail Setup)	:SOURce:PATtern:PON:BLOCK:SET
	Block(Length)	:SOURce:PATtern:PON:BLOCK:LENGth
Output Timing	Burst Mode	:SOURce:PATtern:PON:OTIMing:BMODE
	Burst Cycle	:SOURce:PATtern:PON:OTIMing:BCYCLE
	GT1~GT15	:SOURce:PATtern:PON:OTIMing:GTIMe

Table C-10 Setup:Pattern Panel Command List (3/7)
(200M 16CH PPG/Pattern Mode: PON)

Setup item		Command
PRGM pattern data setup	Character string setting	:SOURce:PATtern:DATA:CHANnel
		:SOURce:PATtern:DA TA:BLOCK
	Binary data setting	:SOURce:PATtern:BDATA:CHANnel
		:SOURce:PATtern:BDATA:BLOCK
PRGM pattern data reversal	CH specification	:SOURce:PATtern:DREVerse:CHANnel
	Block specification	:SOURce:PATtern:DREVerse:BLOCK
	Address specification	:SOURce:PATtern:DREVerse:ADDRes
	Delta specification	:SOURce:PATtern:DREVerse:DELTA
PRGM Pattern Data	Starting/ending address query	:SOURce:PATtern:ADDRes:CHANnel?
		:SOURce:PATtern:ADDRes:BLOCK?

Table C-10 Setup:Pattern Panel Command List (4/7)
(200M 16CH ED/Pattern Mode: Normal)

Setup item		Command
Pattern Input		:SENSe:PATtern[:NORMal]:IMODE
Pattern Setting	Pattern	:SENSe:PATtern[:NORMal]:TYPE
Pattern Setting (PRBS)	Mark Ratio	:SENSe:PATtern[:NORMal]:PRBS:MRATio
	Bit Shift	:SENSe:PATtern[:NORMal]:PRBS:BSHift
Pattern Setting (Zero-Subst)	Pattern Length	:SENSe:PATtern[:NORMal]:ZSUBstitute:LENGth
	Zero-Length	:SENSe:PATtern[:NORMal]:ZSUBstitute:ZLENGth
	Logic	:SENSe:PATtern[:NORMal]:ZSUBstitute:LOGic
Pattern Setting (PRGM)	Pattern Length	:SENSe:PATtern[:NORMal]:PROGram:LENGth
	Logic	:SENSe:PATtern[:NORMal]:PROGram:LOGic
Pattern Setting (MIX)	PRGM(Logic)	:SENSe:PATtern[:NORMal]:MIX:PROGram:LOGic
	PRBS(Pattern)	:SENSe:PATtern[:NORMal]:MIX:PRBS:TYPE
	PRBS(Mark Ratio)	:SENSe:PATtern[:NORMal]:MIX:PRBS:MRATio
	PRBS(Bit Shift)	:SENSe:PATtern[:NORMal]:MIX:PRBS:BSHift
	Block(Row, Column)	:SENSe:PATtern[:NORMal]:MIX:BLOCK:SIZE
	Block(Detail Setup)	:SENSe:PATtern[:NORMal]:MIX:BLOCK:SET
	Block (Column Length)	:SENSe:PATtern[:NORMal]:MIX:BLOCK:LENGth
Sync Setting	Auto Sync(PRGM)	:SENSe:PATtern[:NORMal]:SYNC:ASYNc:PROGram
	Auto Sync(PRBS)	:SENSe:PATtern[:NORMal]:SYNC:ASYNc:PRBS
	Threshold(PRGM)	:SENSe:PATtern[:NORMal]:SYNC :THReshold:PROGram
	Threshold(PRBS)	:SENSe:PATtern[:NORMal]:SYNC:THReshold:PRBS
	Pattern Sync Mode	:SENSe:PATtern[:NORMal]:SYNC:PSMode
	Burst Sync Mode	:SENSe:PATtern[:NORMal]:SYNC:BSMode
	Frame Length	:SENSe:PATtern[:NORMal]:SYNC:FLENGth
Burst Setting	Enable Length	:SENSe:PATtern[:NORMal]:BURSt:ELENGth
	PRGM Data Mode	:SENSe:PATtern[:NORMal]:BURSt:PMODE

Table C-10 Setup:Pattern Panel Command List (5/7)
(200M 16CH ED/Pattern Mode: PON)

Setup item		Command
Enable CHs		:SENSe:PATtern:PON:ECHeannel
OH/Information Pattern	PRGM(Logic)	:SENSe:PATtern:PON:PROGrama:LOGic
	PRGM(Data Mode)	:SENSe:PATtern:PON:PROGrama:MODE
	PRBS(Pattern)	:SENSe:PATtern:PON:PRBS:TYPE
	PRBS(Mark Ratio)	:SENSe:PATtern:PON:PRBS:MRATio
	PRBS(Bit Shift)	:SENSe:PATtern:PON:PRBS:BSHift
	Block (Number of blocks)	:SENSe:PATtern:PON:BLOCK:SIZE
	Block(Detail Setup)	:SENSe:PATtern:PON:BLOCK:SET
	Block(Length)	:SENSe:PATtern:PON:BLOCK:LENGth
Sync Setting	Auto Sync(PRGm)	:SENSe:PATtern:PON:SYNC:ASYNc:PROGrama
	Auto Sync(PRBS)	:SENSe:PATtern:PON:SYNC:ASYNc:PRBS
	Threshold(PRGm)	:SENSe:PATtern:PON:SYNC:THReshold:PROGrama
	Threshold(PRBS)	:SENSe:PATtern:PON:SYNC:THReshold:PRBS
	Burst Sync Mode	:SENSe:PATtern:PON:SYNC:BSMode
	Frame Length	:SENSe:PATtern:PON:SYNC:FLENGth

Table C-10 Setup:Pattern Panel Command List (6/7)
(200M 16CH ED/Pattern Mode: PON)

Setup item		Command
PRGM pattern data setup	Character string setting	:SENSe:PATtern:DATA:CHANnel
		:SENSe:PATtern:DATA:BLOCK
	Binary data setting	:SENSe:PATtern:BDATA:CHANnel
		:SENSe:PATtern:BDATA:BLOCK
PRGM pattern data reversal	CH specification	:SENSe:PATtern:DREVerse:CHANnei
	Block specification	:SENSe:PATtern:DREVerse:BLOCK
	Address specification	:SENSe:PATtern:DREVerse:ADDReSS
	Delta specification	:SENSe:PATtern:DREVerse:DELTA
PRGM Pattern Data query	Starting/ending address	:SENSe:PATtern:ADDReSS:CHANnel?
		:SENSe:PATtern:ADDReSS:BLOCK?

Table C-10 Setup:Pattern Panel Command List (7/7)

Setup item	Command
Display page switching	:DISPlay:SEtup:PATtern:PAGE
Display CH switching	:DISPlay:SEtup:PATtern:CHANnel

- Setup:AUX I/F panel

Table C-11 Setup:AUX I/F Panel Command List

Setup item		Command
AUX Port	Level	:OUTPut:AUXiliary:LEVel
AUX Port Port Setting	Source (CH setting)	:OUTPut:AUXiliary:CHANnel
	Source (Type setting)	:OUTPut:AUXiliary:TYPE
	Logic	:OUTPut:AUXiliary:LOGic
AUX Port Output Timing	Start Location	:OUTPut:AUXiliary:BLOCation
	Stop Location	:OUTPut:AUXiliary:ELOCation
	POS Delay	:OUTPut:AUXiliary:ENVELOpe:PDELay
	NEG Delay	:OUTPut:AUXiliary:ENVELOpe:NDELay
	Pulse Width	:OUTPut:AUXiliary:RESet:PWIDth
	Location	:OUTPut:AUXiliary:RESet:LOCation
Display port switching		:DISPlay:SETup:AUXiliary:PORT

- Setup:Trigger I/F panel

Table C-12 Setup:Trigger I/F Panel Command List (1/2) (200M 16CH PPG)

Setup item			Command
Sync Output	Source(Output signal)		:OUTPut:SYNC:SOURce
	Pattern Sync	CH Select	:OUTPut:SYNC:PATtern:CHANnel
		Source	:OUTPut:SYNC:PATtern:TYPE
Trigger Output	Source		:OUTPut:TRIGger:SOURce
	CH Select		:OUTPut:TRIGger:CHANnel
	Block Select		:OUTPut:TRIGger:BLOCK
Burst Trigger Output	Output Delay		:OUTPut:BTRigger:DELay

Table C-12 Setup:Trigger I/F Panel Command List (2/2) (200M 16CH ED)

Setup item			Command
Sync Output	Source(Output signal)		:OUTPut:SYNC:SOURce
	Pattern Sync	CH Select	:OUTPut:SYNC:PATtern:CHANnel
		Source	:OUTPut:SYNC:PATtern:TYPE
	Clock Sync	CH Select	:OUTPut:SYNC:CLOCK:CHANnel

■ Test Menu Subwindow

Table C-13 Test Menu Subwindow Command List

Setup item	Command
Panel switching	:DISPlay:TEST[:NAME]

● Test Menu:Measurement panel

Table C-14 Test Menu:Measurement Panel Command List (1/6)

Setup item	Command
Test Item	:SENSe:MEASure:TEST :SOURce:PATtern:DElay:TYPE

Table C-14 Test Menu:Measurement Panel Command List (2/6)
(Test Item:Error/Alarm)

Setup item	Command
MEAS.Mode	:SENSe:MEASure:EALarm:MODE
MEAS.Time	:SENSe:MEASure:EALarm:PERiod
Timed Start	:SENSe:MEASure:EALarm:BTIME:SET
ON/OFF	
Start Time	:SENSe:MEASure:EALarm:BTIME:STARt
Error Detect	:SENSe:MEASure:EALarm:ERRor:MODE
Detected Block	:SENSe:MEASure:EALarm:ERRor:DBLock
Block Select	:SENSe:MEASure:EALarm:ERRor:BLOCK
Clock Loss Evaluation	:CALCulate:EALarm:CLEValuation
Sync Loss Evaluation	:CALCulate:EALarm:SLEValuation
EI/%EFI	:CALCulate:EALarm:ERRor:INTerval
Performance	:CALCulate:EALarm:PERFomance:MEASurement
Threshold	:CALCulate:EALarm:PERFomance:THReshold
Threshold EI/%EFI	:CALCulate:EALarm:TEINTerval:MEASurement
Power Fail	:CALCulate:EALarm:PFAil

Table C-14 Test Menu:Measurement Panel Command List (3/6)
(Test Item:Error/Alarm)

Setup item		Command
Buzzer(Error)	CH selection	:SYSTem:BEEPer:ERRor:CHANnel
Buzzer(Alarm)	CH selection	:SYSTem:BEEPer:ALARm:CHANnel
Display CH switching		:DISPlay:TEST:EALarm:CHANnel

Table C-14 Test Menu:Measurement Panel Command List (4/6)
(Test Item: Delay (Mode 1))

Setup item		Command
MEAS. Mode		:SENSe:MEASure:DELAy1:MODE
		:SENSe:MEASure:DELAy1:TYPE
Time Out		:SENSe:MEASure:DELAy1:TOUT
Measument CH		:SENSe:MEASure:DELAy1:CHANnel
Pattern Insertion Cycle		:SOURce:PATTern:DELAy:CYCLE

Table C-14 Test Menu:Measurement Panel Command List (5/6)
(Test Item: Delay (Mode 2))

Setup item		Command
MEAS. Mode		:SENSe:MEASure:DELAy2:MODE
		:SENSe:MEASure:DELAy2:TYPE
Time Out		:SENSe:MEASure:DELAy2:TOUT
Detected Pattern		:SENSe:MEASure:DELAy2:PATTern
Measurement Unit		:SENSe:MEASure:DELAy2:SET
Measument CH		:SENSe:MEASure:DELAy2:CHANnel

Table C-14 Test Menu:Measurement Panel Command List (6/6)
(Test Item: Eye Margin)

Setup item		Command
MEAS.Mode		:SENSe:MEASure:EMARgin:MODE
		:SENSe:MEASure:EMARgin:TYPE
Error Threshold		:SENSe:MEASure:EMARgin:MARGin:THReshold
		:SENSe:MEASure:EMARgin:MARGin:RESolutio
Threshold Select		:SENSe:MEASure:EMARgin:DIAGram:THReshold
MEAS. Unit		:SENSe:MEASure:EMARgin:UNIT
MEAS. CH		:SENSe:MEASure:EMARgin:CHANnel

- Test Menu:Error Addition Panel

Table C-15 Test Menu:Error Addition Panel Command List

Setup item	Command
Error Addition	:SOURce:PATtern:EADDITION:SET
Single	:SOURce:PATtern:EADDITION:SINGLE
Mode	:SOURce:PATtern:EADDITION:MODE1 :SOURce:PATtern:EADDITION:MODE2 :SOURce:PATtern:EADDITION:MODE3
Rate	:SOURce:PATtern:EADDITION:RATE
Burst Cycle	:SOURce:PATtern:EADDITION:BCYCLE
Enable Length	:SOURce:PATtern:EADDITION:ELENGth
Addition	:SOURce:PATtern:EADDITION:ADDITION
Additional Block	:SOURce:PATtern:EADDITION:ABLOCK
Select Block	:SOURce:PATtern:EADDITION:BLOCK
Display CH switching	:DISPlay:TEST:EADDITION:CHANnel

- Test Menu:Analyze panel

Table C-16 Test Menu:Analyze Panel Command List

Setup item	Command
Select Analyze CH	:SENSe:ANALyze:CHANnel
Tract Text data selection	:SENSe:ANALyze:TTEXT:DATA
Trace Text (Intermediate Data)	Detail selection :SENSe:ANALyze:TTEXT:IDATa:ITEM Interval Time :SENSe:ANALyze:TTEXT:IDATa:PERiod
Trace Text(End Data)	:SENSe:ANALyze:TTEXT:EDATa:ITEM
1s Data(Threshold)	:SENSe:ANALyze:TTEXT:SDATa:THReshold

■ Result Subwindow (Test Item common)

Table C-17 Result Subwindow Command List

Setup item		Command
Auto Search	Mode	:SENSe:MEASure:ASEarch:MODE
	CH Select	:SENSe:MEASure:ASEarch:CHANnel
	PRBS Pattern	:SENSe:MEASure:ASEarch:PATTern
	Search	
	Start	:SENSe:MEASure:ASEarch:STARt
	Stop	:SENSe:MEASure:ASEarch:STOP
	Status query	:SENSe:MEASure:ASEarch:STATe?
	Failing CH query	:SENSe:MEASure:ASEarch:FCHannel?
Alarm monitor		:CALCulate:DATA:MONitor?
Start		:SENSe:MEASure:STARt
Stop		:SENSe:MEASure:STOP
Time(Measurement time display mode)		:DISPlay:RESult:TIME
Measurement data buffering		:CALCulate:DATA:STORE
Measurement data storage buffer clear		:CALCulate:DATA:CLEar

■ Result subwindow (Test Item:Error/Alarm)

Table C-18 Result Subwindow (Test Item:Error/Alarm) Command List

Setup item	Command
Measurement start time query	:SENSe:MEASure:EALarm:STARt?
Measurement end time query	:SENSe:MEASure:EALarm:STOP?
Measurement status query	:SENSe:MEASure:EALarm:STATe?
Elapsed time query	:SENSe:MEASure:EALarm:ELAPsed?
Remaining time query	:SENSe:MEASure:EALarm:TIMed?
Intermediate data creation time query	:SENSe:MEASure:EALarm:ITIME?
Measurement result query	:CALCulate:DATA:EALarm?
Panel switching	:DISPlay:RESult:EALarm[:NAME]
Display	:DISPlay:RESult:EALarm:MODE
Unit	:DISPlay:RESult:EALarm:UNIT

- **Result:ALL panel**

Table C-19 Result ALL Panel Command List

Setup item		Command
Display	Error measurement	:DISPlay:RESult:EALarm:TYPE
	item selection	
	Performance Item	:DISPlay:RESult:EALarm:PITem
	Threshold	:DISPlay:RESult:EALarm:TITem
	EI/%EFI Item	

- **Result:Normal panel**

Table C-20 Result:Normal Panel Command List

Setup item	Command
Display CH switching	:DISPlay:RESult:EALarm:CHANnel <ch>

- **Result:Zoom panel**

Table C-21 Result:Zoom Panel Command List

Setup item		Command
Display1	Unit	:DISPlay:RESult:EALarm:ZOOM1:UNIT
	CH	:DISPlay:RESult:EALarm:ZOOM1:CHANnel
	Item	:DISPlay:RESult:EALarm:ZOOM1:ITEM
	Performance Item	:DISPlay:RESult:EALarm:ZOOM1:PITem
	Threshold	:DISPlay:RESult:EALarm:ZOOM1:TITem
	EI/%EI Item	
Display2	Unit	:DISPlay:RESult:EALarm:ZOOM2:UNIT
	CH	:DISPlay:RESult:EALarm:ZOOM2:CHANnel
	Item	:DISPlay:RESult:EALarm:ZOOM2:ITEM
	Performance Item	:DISPlay:RESult:EALarm:ZOOM2:PITem
	Threshold	:DISPlay:RESult:EALarm:ZOOM2:TITem
	EI/%EI Item	
Display3	Unit	:DISPlay:RESult:EALarm:ZOOM3:UNIT
	CH	:DISPlay:RESult:EALarm:ZOOM3:CHANnel
	Item	:DISPlay:RESult:EALarm:ZOOM3:ITEM
	Performance Item	:DISPlay:RESult:EALarm:ZOOM3:PITem
	Threshold	:DISPlay:RESult:EALarm:ZOOM3:TITem
	EI/%EI Item	
Display4	Unit	:DISPlay:RESult:EALarm:ZOOM4:UNIT
	CH	:DISPlay:RESult:EALarm:ZOOM4:CHANnel
	Item	:DISPlay:RESult:EALarm:ZOOM4:ITEM
	Performance Item	:DISPlay:RESult:EALarm:ZOOM4:PITem
	Threshold	:DISPlay:RESult:EALarm:ZOOM4:TITem
	EI/%EI Item	

- **Result:Monitor panel**

Table C-22 Result:Monitor Panel Command List

Setup item	Command
History Reset	:DISPlay:RESult:EALarm:HRESet

- **Result subwindow (Test Item:Delay (Mode 1))**

Table C-23 Result Subwindow (Delay (Mode 1)) Command List

Setup item	Command
Measurement start time query	:SENSe:MEASure:DELAy1:STARt?
Measurement end time query	:SENSe:MEASure:DELAy1:STOP?
Measurement status query	:SENSe:MEASure:DELAy1:STATe?
Measurement progress query	:SENSe:MEASure:DELAy1:MSTate?
Elapsed time query	:SENSe:MEASure:DELAy1:ELAPsed?
Measurement result query	:CALCulate:DATA:DELAy1?

- **Result subwindow (Test Item:Delay (Mode 2))**

Table C-24 Result Subwindow (Delay (Mode 2)) Command List

Setup item	Command
Measurement start time query	:SENSe:MEASure:DELAy2:STARt?
Measurement end time query	:SENSe:MEASure:DELAy2:STOP?
Measurement status query	:SENSe:MEASure:DELAy2:STATe?
Measurement progress query	:SENSe:MEASure:DELAy2:MSTate?
Elapsed time query	:SENSe:MEASure:DELAy2:ELAPsed?
Measurement result query	:CALCulate:DATA:DELAy2?

■ Result subwindow (Test Item:Eye Margin)

Table C-25 Result Subwindow (Test Item:Eye Margin) Command List

Setup item		Command
Measurement start time query		:SENSe:MEASure:EMARgin:STARt?
Measurement end time query		:SENSe:MEASure:EMARgin:STOP?
Measurement status query		:SENSe:MEASure:EMARgin:STATe?
Elapsed time query		:SENSe:MEASure:EMARgin:ELAPsed?
Diagram valid point count query		:SENSe:MEASure:EMARgin:DIAGram:POINT?
Recall Diagram Measurement start time query		:SENSe:MEASure:RDIagram:STARt?
Recall Diagram measurement end time query		:SENSe:MEASure:RDIagram:STOP?
Recall Diagram valid point count query		:SENSe:MEASure:RDIagram:POINT?
Recall Diagram measurement CH query		:SENSe:MEASure:RDIagram:CHANnel?
Recall Diagram measurement Error Rate query		:SENSe:MEASure:RDIagram:THREshold?
Measurement result query		:CALCulate:DATA:EMARgin?
Display		:DISPlay:RESult:EMARgin:ERATe
Scale	Auto Scale	:DISPlay:RESult:EMARgin:SCALE:ASCale
	Threshold Voltage	:DISPlay:RESult:EMARgin:SCALE:VOLTage
	Phase	:DISPlay:RESult:EMARgin:SCALE:PHASe
Marker,△Marker	Display ON/OFF status	:DISPlay:RESult:EMARgin:MARKer:SET
	Movement mode	:DISPlay:RESult:EMARgin:MARKer:MODE
	Object of movement	:DISPlay:RESult:EMARgin:MARKer:ERATe
	Movement	:DISPlay:RESult:EMARgin:MARKer:MOVE
	Position query	:DISPlay:RESult:EMARgin:MARKer:POSition?

■ Analyze subwindow

Table C-26 Analyze Subwindow Command List

Setup item		Command
Mode		:DISPlay:ANALyze:MODE
Unit		:DISPlay:ANALyze:UNIT
Display CH		:DISPlay:ANALyze:CHANnel
Trace Text	Data query	:CALCulate:DATA:TEXT?
Recall Text	Scrolling	:DISPlay:ANALyze:TEXT:SCRoll
	Time	:DISPlay:ANALyze:TEXT:TIME
Data start time query		:CALCulate:DATA:START?
Data size query		:CALCulate:DATA:SIZE?

■ Customize subwindow

Table C-27 Customize Subwindow Command List

Setup item		Command
Setup1	Unit	:DISPlay:CUSTmize:SETup1:UNIT
	CH Select	:DISPlay:CUSTmize:SETup1:CHANnel
	Item	:DISPlay:CUSTmize:SETup1:ITEM
Setup2	Unit	:DISPlay:CUSTmize:SETup2:UNIT
	CH Select	:DISPlay:CUSTmize:SETup2:CHANnel
	Item	:DISPlay:CUSTmize:SETup2:ITEM
Setup3	Unit	:DISPlay:CUSTmize:SETup3:UNIT
	CH Select	:DISPlay:CUSTmize:SETup3:CHANnel
	Item	:DISPlay:CUSTmize:SETup3:ITEM
Setup4	Unit	:DISPlay:CUSTmize:SETup4:UNIT
	CH Select	:DISPlay:CUSTmize:SETup4:CHANnel
	Item	:DISPlay:CUSTmize:SETup4:ITEM
Setup5	Unit	:DISPlay:CUSTmize:SETup5:UNIT
	CH Select	:DISPlay:CUSTmize:SETup5:CHANnel
	Item	:DISPlay:CUSTmize:SETup5:ITEM
Setup6	Unit	:DISPlay:CUSTmize:SETup6:UNIT
	CH Select	:DISPlay:CUSTmize:SETup6:CHANnel
	Item	:DISPlay:CUSTmize:SETup6:ITEM
Pattern(Unit)	Unit	:DISPlay:CUSTmize:PATtern:UNIT
	CH Select	:DISPlay:CUSTmize:PATtern:CHANnel
	Offset	:DISPlay:CUSTmize:PATtern:OFFSet
	Block	:DISPlay:CUSTmize:PATtern:BLOCK
Result	Unit	:DISPlay:CUSTmize:RESult:UNIT
	CH Select	:DISPlay:CUSTmize:RESult:CHANnel
	Item	:DISPlay:CUSTmize:RESult:ITEM

■ Other commands

Table C-28 Other Commands

Setup item	Command
Device I/O buffer size query	:SYSTem:BSIZE?
Error status query	:SYSTem:ERRor?
SCPI version query	:SYSTem:VERSion?
Status register setup/query	STATus subsystem

